Center for Technology for Advanced Scientific Component Software (TASCS)

Project Management Plan

Version 0.91 *(Draft)*
5 December 2006

Programmatic Information:

Proposal Number:	100990
Program Announcements:	LAB 06-04 and DE-FG02-06ER06-04
Program Area:	Center for Enabling Technology
Program Office:	Office of Advanced Scientific Computing Research
Technical Contact:	Frederick C. Johnson

Lead Principal Investigator:

Institution	*Principal Investigator*
Oak Ridge National Laboratory	David E. Bernholdt
PO Box 2008, MS 6016	(865) 574-3147
Oak Ridge, TN 37831–6016	bernholdtde@ornl.gov

Participating Institutions

Institution	*Institutional Lead co-PI*
Argonne National Laboratory (ANL)	Lois Curfman McInnes
Binghamton University (BU)	Madhusudhan Govindaraju
Indiana University (IU)	Randall Bramley
Lawrence Livermore National Laboratory (LLNL)	Gary Kumfert
Los Alamos National Laboratory (LANL)	Craig Rasmussen
Oak Ridge National Laboratory (ORNL)	James A. Kohl
Pacific Northwest National Laboratory (PNNL)	Jarek Nieplocha
Sandia National Laboratories (SNL)	Rob Armstrong
Tech-X Corporation (Tech-X)	Svetlana Shasharina
University of Maryland (UMD)	Alan Sussman
University of Utah (UU)	Steven G. Parker

Contents

1 Roadmap

This document lays out the management plan for the Center for Technology for Advanced Scientific Component Software (TASCS), as requested by our sponsor, the Department of Energy (DOE) Office of Science. We begin with a brief technical overview of the project (Section 2) to help the reader understand our management structure (Section 3), the baseline from which TASCS begins (Section 8), our milestones (Section 9), and roles of TASCS participants, both at the institutional and individual levels (Section 10). We also discuss our plans for interacting with Scientific Discovery through Advanced Computing (SciDAC) applications and the broader community (Section 4), reporting to our sponsors (Section 5), the software lifecycle (Section 6), and our approach to intellectual property issues (Section 7).

2 Technical Overview

TASCS is part of a long-term effort with the far-reaching goal of fundamentally changing the way scientific software is developed and used. At the heart of this effort is the concept of component-based software engineering (CBSE), which is already well-established in other mainstream areas of computing as a key means of managing the burgeoning complexity of large-scale software systems. CBSE combines object-oriented design with the powerful features of well-defined common interfaces, programming language interoperability, and dynamic composability. The grass-roots Common Component Architecture (CCA) Forum was founded in 1998 on the belief that a similar approach could address the looming complexity crisis in high-performance scientific simulations, which have become central to new scientific discovery in the DOE and elsewhere. Unlike commercial projects, the CCA Forum addresses the challenges associated with maintaining high performance, working with a broad spectrum of programming languages and computer architectures, and helping to preserve DOE investments in legacy codes.

Prior work, led primarily by the the SciDAC-1 Center for Component Technology for Terascale Simulation Software (CCTTSS), has prototyped the concepts and implementation of CBSE for high-performance scientific computing, in the form of the CCA (see Section 8). TASCS extends this work with the goals of bringing the current prototype-level capabilities to production quality and capitalizing on the dynamic nature of components to address challenges in petascale scientific simulations.

Technically, TASCS is organized into four central thrust areas, each with several activities. Thrusts and activities have designated technical leads, who are responsible for coordinating the activities of all participants. An overview of corresponding milestones and deliverables can be found in Section 9.

- **Component Technology Initiatives** [Coordinator: Lois Curfman McInnes, ANL] focus on utilizing and extending the component model to provide new "value added" capabilities for CCA users.
  - **Emerging high-performance computing (HPC) Hardware and Software Paradigms** [Coordinator: Jarek Nieplocha, PNNL] focuses on developing component-based tools to help applications manage higher/hybrid levels of parallelism through a multiple-component multiple-data (MCMD) paradigm; this work includes support for hardware co-processors such as Field-Programmable Gate Array (FPGA)s.
  - **Software Quality and Verification** [Coordinator: Tammy Dahlgren, LLNL] focuses on mechanisms to specify and verify functional software "contracts" associated with component interfaces to help developers and users improve the quality of their code.
  - **Computational Quality of Service and Adaptivity** [Coordinator: Lois Curfman McInnes, ANL] focuses on mechanisms to dynamically adapt long-running component-based applications in response to changing conditions (i.e., performance, accuracy, mathematical consistency, reliability, etc.) by composing, substituting, and reconfiguring components on the fly.

- The **CCA Environment** [Coordinator: Gary Kumfert, LLNL] thrust supports and improves the foundation of the CCA environment and tools for ease of use and as core technology for other initiatives.
  - **Core Tool Support and Maintenance** [Coordinator: Ben Allan, SNL] provides essential software maintenance, porting, and support in the face of changing HPC environments.
  - **Enhancements** [Coordinator: Tom Epperly, LLNL] extend the CCA environment with additional features/capabilities required by users and by other activities within TASCS.
  - **Usability** [Coordinator: Craig Rasmussen, LANL] focuses on making CCA technology more accessible to users through the development of a tiered approach ("CCA Lite") and tools for debugging and testing components.
- The **CCA Toolkit** [Coordinator: Rob Armstrong, SNL] will provide a diverse suite of scientific components, along with a basic software skeleton to facilitate the creation of new components.
- **User and Application Outreach and Support** [Coordinator: David Bernholdt, ORNL] focuses on assisting applications groups with CCA adoption and usage through direct interactions, and development of documentation, tutorials, and example materials.

3 Project Management and Internal Communication

The project team consists of eleven institutions: six national laboratories (ANL, LANL, LLNL, ORNL, PNNL, and SNL), four universities (BU, IU, UMD, and UU), and a research-based company (Tech-X). Eight of the organizations constituted the team for the earlier SciDAC CCTTSS project (2001–2006). The three new additions to the team (BU, Tech-X, and UMD) have been active participants in the CCA Forum for several years and have had strong collaborative ties with the CCTTSS. Our approach to managing the project draws heavily on the experience of the CCTTSS project, which we found to be very effective.

The work of the project is structured to be well integrated and highly collaborative across the team members. Each of the activities, at the level outlined in Section 2, involves between two and eight of the eleven institutions in TASCS. The majority of the individual milestones and deliverables described in Section 9 will be carried out by small multi-institutional teams.

Our approach to managing the project recognizes both its administrative and technical aspects. Each institution has a lead co-PI, who is responsible for administrative and site-specific matters. As described in Section 2, technical areas also have designated leaders, who are responsible for overseeing and coordinating the scientific work of the project. (There is significant overlap between these two groups: six of the nine technical leads are also institutional co-PIs.)

The lead PI (Bernholdt) oversees all aspects of the project, both administrative and technical, working with the site and technical leads as appropriate to carry out the work of the project. Decisions will be made by consensus of the leadership, with the lead PI acting as arbiter and final decisionmaker. The lead PI will also serve as the primary point of contact with DOE management.

Though our team is large, we have a history of effective and productive collaboration stretching back, in most cases, seven or more years, at both the institutional and individual levels. As a consequence of our experience together, project participants are very much accustomed to remote collaboration (as both a general approach and working with this particular group of people).

Mailing lists and individual email provide the primary means of communication among project members. We have private mailing lists for the TASCS leadership and for all TASCS participants, as well as a variety of topical mailing lists aligned with technical elements of the project.

Teleconferences, often supplemented by the MeetingPlace web sharing service provided by ESnet, provide synchronous communications when needed. Most project members have easy access to teleconferencing services, and several participants have dedicated teleconference lines. We have a regularly scheduled time slot for coordination meetings of the leadership team, which we anticipate holding every 2–3 weeks, as needed. Other subgroups hold teleconferences as dictated by the needs of the project.

The mailing list capabilities used by TASCS are part of the suite of services offered by the *cca-forum.org* collaboration server, which the CCTTSS and now TASCS operates for the benefit of both the project and the larger CCA community. This server also provides web and wiki services, revision control repositories, bug trackers, and other tools for collaborative software development. These capabilities are used on a daily basis by the TASCS project.

In addition to the broad spectrum of resources for remote collaboration used by the project, we also have face-to-face meetings on a fairly frequent basis. The CCA Forum's regular quarterly meetings serve many of the purposes of project meetings due to the large overlap of Forum members and TASCS participants. Forum meetings typically include reports of significant results from work in progress, discussion of the CCA specification and environment, and other relevant topics.

In addition to these activities, which are very free-form (the agenda is dynamic and varies every meeting), we have recently instituted the practice of having TASCS activity leads present brief progress reports at every Forum meeting to increase awareness of the various activities both within TASCS and in the larger CCA community.

CCA Forum meetings are normally 1.5 days in length, but it is common for individual working groups to get together for an additional half-day or full-day meeting to allow for more in-depth work. Although primarily intended as a face-to-face meeting, CCTTSS and now TASCS do provide remote access to the meetings through teleconferencing and MeetingPlace web sharing. By holding progress reports at the Forum meetings and regular coordination teleconferences in between, we hope to improve the management of TASCS over CCTTSS and insure the center's responsiveness to the SciDAC-2 program as a whole.

Finally, the most important asset to managing TASCS is the commitment to community that the members feel. When a hole in the CCA software portfolio is identified, or an opportunity to engage potential users arises, members often step up and volunteer without being asked. It is this commitment to community that makes managing TASCS viable.

4 Interactions with Applications Partners and the Larger Community

We have established three different mechanisms to support our work with applications. First, we have built support into our Component Technology Initiatives for the necessary level of interaction for targeted applications to insure that TASCS goals are achieved. Second are projects for which we have joint funding, either in the form of Scientific Application Pilot (SAP) projects or through "embedding" of CCA-related personnel and activities in the application proposal. Third, we have allocated a modest level of effort to provide support for other projects, which we often refer to as "walk-up" applications.

During the development of the proposal, we identified a sizable number of potential collaborations – many represent continuations of collaborations established during the CCTTSS project. One of our initial tasks, currently in progress, is to contact these projects to determine their funding status, and whether CCA-related activities are still in their planned scope of work. Our current list of prospective applications can be found in Appendix A.

Prioritization among opportunities will be based on the level of support required, the likelihood of contributing towards significant advances on the scientific side, and the prospects of obtaining new experience and knowledge of value to the wider CCA community.

Drawing on our experience in CCTTSS, our standard practice in interacting with all collaborating projects will be to clearly designate a point of contact within TASCS who has the primary responsibility for coordinating as a liaison between the two projects. Because many of our collaborations involve individuals who participate in *both* TASCS and the partner project, the communication challenge is much simpler than it might be otherwise. Where there is not a single individual who can serve as the point of contact for both sides, we designate one on the TASCS side and ask the collaborating project to do the same. In a management sense, the points of contact with collaborating projects are responsible to the appropriate thrust area lead, depending on the nature of the collaboration, and ultimately to the lead PI.

While collaborative interactions with outside projects tend to be more individual, and are primarily coordinated by the point of contact, TASCS as a whole can provide a number of services meant to support such collaborations. Where individual educational outreach is not sufficient (for example because of desire to acquaint an entire project with CBSE and the CCA), we can provide more formal tutorials to specialized audiences. We can also convene “coding camps”, which are intensive working meetings, typically 3–5 days in duration, bringing members of a project team or scientific community together with CCA developers and experts. Participants come to the camp with certain CCA-related goals, defined as part of the planning process, and work together during the camp to realize them. Convening users and experts in the same physical location for an extended period allows for much quicker development cycles and problem solving than the more typical asynchronous interactions that occur when everyone is at their home institutions. In the CCTTSS project, we found coding camps to be a very effective way of jumpstarting a collaboration or getting over complex hurdles. In addition, TASCS makes all the same collaboration tools described in Section 3 available to our partners as well, for use at their convenience.

TASCS also participates in a number of other, broader communities in a more general way. The CCA Forum includes many participants interested in the Common Component Architecture – both developers and users, and with a variety of different funding sources. Because of its history and its size, TASCS serves as the nucleus of the Forum community and operates very openly within it. Continuing the practice of the CCTTSS, TASCS provides the hardware for and operates the Forum’s collaboration server. Reaching beyond the CCA Forum, TASCS is working to foster the development of a larger international community of developers of high performance computational frameworks and component technology, by taking a leading role in the organization of an annual “CompFrame” workshop, Birds of a Feather sessions at Supercomputing, and mailing lists. TASCS also engages the computational science community through the organization of other workshops, minisymposia, talks, tutorials, and papers.

During SciDAC-1, the primary means of interaction with other portions of the SciDAC community, with whom we did not have established collaborations, was at various meetings, including the annual SciDAC PI meetings. We expect this to continue in TASCS, but we also hope that DOE will further help to establish formal mailing lists that will make it easier to communicate with the PIs of other SciDAC projects, perhaps through the SciDAC Outreach Center. This would be valuable for announcements of various tutorials, workshops and other events of interest, as well as for general questions and information requests.

The Outreach Center is a new feature of the SciDAC program, and as such it will take some time and experimentation to understand how best to work with it and capitalize on it. Based on preliminary conversations with David Skinner, the Center’s PI, it appears that the Outreach Center will serve as a central clearinghouse and “matchmaker” service for the SciDAC program, and will provide consulting and support services for software engineering. On our part, in TASCS, this will entail some education for Outreach Center personnel as to the nature of our work and our software products. Since the CCA is also intimately associated with the software development and engineering practices of several other groups that use it, we plan to work closely with the Outreach Center to align our approaches, so that the integration of CCA tools into user applications will as closely as possible adhere to the “best practices” and use the tools that the Outreach Center will recommend and support.

Finally, TASCS helps educate the next generation of computer and computational scientists in advanced software technologies, like the CCA, through the inclusion of CCA-related material in courses taught at all four participating universities. At least five courses now include CCA technology and ideas, and students from the four universities have done summer internships with TASCS laboratories.

5 Reporting

TASCS will provide progress reports to DOE at six month intervals. These reports will include contributions from all participating institutions. They will provide highlights summarizing recent work, discuss progress, track institutional spending, draw attention to issues, and make recommendations related to both the TASCS project, and the larger SciDAC program, as appropriate.

6 Software Support and Maintenance Plan

Software is one of the primary products of the TASCS activity, and support for this product is vital to the long term impact of this effort. We have recognized the importance of the software that implements the "CCA environment" by casting its support and enhancement as one of the four central thrusts of the TASCS project.

The current TASCS software base, inherited from the CCTTSS project, uses basic software engineering tools practices, such as revision control repositories and bug trackers. Historically, the various tools are products primarily of teams at specific institutions. However, development is open, and developers from other institutions routinely contribute enhancements and bug fixes to the various tools. While the level of testing associated with different software products currently varies significantly, we plan to establish continuous integration testing across the software suite, which will help to improve the overall quality of the released software.

Maintenance and support of the the TASCS software includes coordinating bug fixes, supporting users, and porting and tuning for Leadership Class and other computer platforms important for our users. As part of TASCS, we recently established a centralized "help desk" as a point of contact for all user issues, which, once diagnosed, can be assigned to the specific tools.

A significant software challenge faced by TASCS has to do with "build systems". The standard Unix/Linux build tools are widely recognized as being inadequate to the meet the needs of today's complex software systems and their interdependencies, but unfortunately, there are no better solutions in wide use. This situation affects the building of many HPC scientific codes, including the CCA tools themselves, and becomes even more complicated on current Leadership platforms. We are working within TASCS to improve the user's build experience for our own software as best we can, but a broader-based effort to develop build systems better suited to modern high-end environments could benefit TASCS, as well as many other projects.

Long term (beyond the end of SciDAC-2) support and maintenance of the program's software products poses some interesting questions. TASCS envisions two categories of software products: the *tools* that implement the CCA-compliant environment and help CCA users develop components, and *components*, which are useful in assembling CCA-based applications. While we plan to make our software products available under open source licenses wherever possible, this is not sufficient to guarantee that the software can be community supported. The CCA tools, in particular, are relatively complex and specialized pieces of software with correspondingly small groups of developers, most of whom are part of TASCS. Support for TASCS-developed components can be broken down into several categories.

- In many cases, TASCS will (assist in) componentize and existing library or tool. Our preference in such cases will be to donate the "componentization" back to the original library developers and let them provide the primary support to the community.

- A secondary strategy will be to work with major users of the components so that they can self-support their own uses of the components, and incidentally provide some level of support to the larger community.

- Support for components by TASCS itself is fallback strategy. Given limited resources, we will have to prioritize our support of the various componentizations based on their value to the project and community, and the complexity of support for the particular package.

As the project progresses, we will continue to analyze the situation and make recommendations as to the most effective approaches and levels of effort needed to continue providing appropriate maintenance and support for TASCS software products.

7 Intellectual Property Plan

As an enabling technology project, TASCS presents an interesting and challenging environment for intellectual property considerations. We can first divide the environment into two parts:

- the CCA specification and software tools which *implement* the CCA environment, and
- the (application) software that *users* of the CCA create, with the help of the aforementioned tools. Such software may be created by the TASCS project as well as by outside groups.

We view our work as primarily a research effort, and generally adhere to the philosophy that software used in research should be openly available to facilitate peer review. We also believe that one of the best ways to form a larger community around an idea like CBSE for high-performance scientific computing is to make the specifications and tools readily accessible to all interested parties. This is consistent with DOE's guidance that the software products of the SciDAC program should be released under an open source license, and that is how we intend to handle the licensing of software under our control.

As an enabling technology, we distinguish the *tools* we create from the *application software* resulting from their use. The vast majority of the application software created with our tools is outside of our control, and it is unwise and impractical for us (or our tools) to dictate the terms under which it might be licensed. Our approach in this context is that the licenses under which our tools are released should not limit the licensing options available to application software created merely through the *use* of those tools.

In addition, we anticipate that a significant fraction of the component software produced with the help of our tools will be based on pre-existing libraries and tools, which will typically already have licenses attached. If the original software is not under an open source license, it is unlikely that componentized version can be (entirely) open source either.

We point out this issue not because we believe it is a problem for TASCS, but rather because we feel it is important to be clear about what elements of the component software ecosystem we do and do not control. Given that our work plan includes the establishment of a repository of components that will be open to contributions, and the componentization of existing software libraries, it is likely that TASCS will be involved in the distribution, and even the creation, of some software that is not open source.

To further complicate this situation, TASCS inherits a significant software base from the preceeding CCTTSS, and in many cases even before that. This software has been licensed based on institutional preferences rather than as a unified project. Existing software that will maintained and developed under TASCS is currently licensed under the following licenses, which we believe all fit the open source criteria:

- BSD derivatives
- GNU Public License (GPL) and derivatives
- Lesser (Library) GNU Public License (LGPL)
- Institution-specific
- Simple copyright assertions

Moreover, these tools rely on third-party software released under various licenses on the list above, as well as the Apache Software License.

While this diversity may be somewhat confusing, we do not see it as a problem. All tools are open source, and the chosen licenses to not impose licensing constraints on application software developed with their assistance. The differences in licenses come into play only for those who wish to modify the tools themselves – a much smaller set of people, to whom we can easily explain the situation. Currently, nearly all contributors to the tools are members of the TASCS project.

So far, no (potential) contributor to the development of CCA software has expressed concern over the current licensing situation, and consequently, we do not see a need to change our approach at this time. However we will continue to discuss intellectual property issues with our contributors and users, with the expectation of making recommendations to DOE as the project evolves.

While most of TASCS products are software, we also produce tutorials and other written materials. For the existing CCA tutorial materials, we have used a Creative Commons "Attribution" license. We plan to continue using similar licenses on our written materials wherever possible.

8 Project Baseline

TASCS picks up where the former SciDAC-1 CCTTSS project left off, building on the groundwork laid by this effort and other contributions by the CCA Forum. At the end of the CCTTSS project, we can broadly say that the concept and implementation of components and CBSE for high-performance scientific computing have successfully reached the prototype stage, and that the job of TASCS is to bring these to a more practical production level.

More specifically, we would characterize the technological baseline, from which TASCS begins its efforts, in the following terms:

- The fundamental characteristics and features of a component architecture for high-performance scientific computing have been captured in the Common Component Architecture (CCA). The CCA has a formal specification process, under the control and oversight of the CCA Forum. The current version of the CCA specification is 0.7, and is considered to be roughly 80% complete with respect to the needs of the mainstream scientific computing community.

- A suite of tools implementing an environment consistent with the CCA specification has been developed and released to the community. These tools provide language interoperability and an HPC-oriented component framework. The implementations thus far exhibit good functionality and moderate usability, and they are generally considered to be in the "beta" stage of development. Additional preliminary tools (frameworks, software development environments) have been developed, but are not widely distributed or used, and would be considered to be more at the "alpha" stage.

Tool	Version	Purpose	Maturity
Babel	1.0	Language interoperability tool (C, C++, Fortran77, Fortran90/95, Java, Python)	Production
Ccaffeine	0.8	HPC parallel component framework	Beta
ccafe-gui	0.8	"Visual programming" interface for the Ccaffeine framework	Beta
Chasm	1.3	Fortran90 array interoperability and semi-automatic interface extraction	Beta (F90 array interop.)/Alpha (interface extraction)
cca-tools	0.6	Integrated software distribution of the tools above and the CCA specification	Alpha
Legion-CCA	1.0α	Component environment built on top of the Legion distributed computing system	Alpha

(continued)

(continued)

Tool	Version	Purpose	Maturity
SCIRun2	0.90	Component framework for MPI, threaded parallel, and distributed environments	Alpha
XCAT	1.5α	Distributed component framework	Alpha

- Through work with a number of scientific applications groups, the general efficacy and potential of the CBSE approach in scientific computing has been established at an anecdotal level. CCA tools and ideas have allowed some users to take novel approaches to the development of their scientific software, which they otherwise would not have pursued without CCA. The following table lists the applications groups we have had the most interaction with prior to the beginning of TASCS, grouped according to their current mode of use of the CCA.

Application or Project	Point of Contact	Scientific Domain
	In Use	
Computational Facility for Reacting Flow Science (CFRFS)	Jaideep Ray, SNL	Combustion
Computational Models for Electron Effects (CMEE)	Peter Stoltz, Tech-X	Electron effects in plasmas
Cooperative Programming	John May, LLNL	Parallel programming models
DistComp	Nanbor Wang, Tech-X	Distributed computing
hypre	Jeff Painter, LLNL	Solvers
Massively Parallel Quantum Chemistry (MPQC)	Curtis Janssen, SNL	Chemistry
NWChem	Theresa Windus, Iowa State U.	Chemistry
Petascale Simulation Initiative (PSI)	Nathan Barton, LLNL	Materials science
ROSE	Dan Quinlan, LLNL	Code refactoring
Tuning and Analysis Utilities (TAU)	Allen Malony, U. Oregon	Performance tools
Terascale Optimal PDE Solvers (TOPS)	Barry Smith, ANL	Solvers
Not reported	M. Diaz, U. Malaga, Spain	Nuclear power plant simulation
	Under Development	
Beam-SBIR	Douglas Dechow, Tech-X	Accelerator beam dynamics
GAMESS-CCA	Masha Sosonkina, Ames Lab	Chemistry
SPARSKIT-CCA	Masha Sosonkina, Ames Lab	Sparse linear algebra
Terascale Tools and Technologies (TSTT)	Lori Diachin, LLNL	Meshing
	Under Evaluation	
eMiriad	Atholl Kemball, U. Illinois	Radio astronomy
Virtual Microbial Cell Simulation (VMCS)	Harold Trease, PNNL	Cell biology
	Influence on Software Design	
Earth System Modeling Framework (ESMF)	Cecelia DeLuca, NCAR	Climate modeling
Terascale Supernova Initiative (TSI)	Doug Swesty, SUNY-Stony Brook	Astrophysics

- Numerous authors, both within and outside CCTTSS have committed to provide components to the CCA Toolkit. Prototyping was done for a common packaging and build environment for Toolkit components, but was not yet ready for release.

- The CCA has been firmly established as the leading approach to CBSE within the DOE HPC computational science community, and to a significant extent in the larger international scientific computing community.

The primary target platforms for the work done to date have been classic "cluster"-style parallel systems running some version of Linux or unix, with commodity Linux clusters as the most common environment in practice. Though this architectural model (the tools designed to it) maps straightforwardly onto other HPC architectures, prior to TASCS, no special consideration has been given to exploitation of features such as multi-processor (shared memory) or multi-core nodes, globally addressable memory, or heterogeneous processors. Likewise, prior work has targeted mainstream, full-featured unix/Linux operating system environments.

9 Overview of Milestones and Deliverables

The following tables provide an overview of the milestones and deliverables of the project, broken down by thrust area and activity. In each section, the coordinator and the participating institutions are listed. These tables should be consistent with the instutional milestones and deliverables provided at the time of the award, and with the project's technical starting point, described in Section 8.

Table 3: **Summary of Milestones for Component Technology Initiatives**
Coordinator: L.C. McInnes, ANL; *Participants:* ANL, IU, LLNL, ORNL, PNNL, SNL, UMD

Year 1	*Year 2*	*Year 3*	*Years 4–5*
Emerging HPC Paradigms *Coordinator:* J. Nieplocha, PNNL; *Participants:* ORNL, PNNL *Motivating Applications:* biology and quantum chemistry simulations			
• Develop multi-level parallelism model. • Define abstract model for CCA hybrid apps.	• Develop CCA model for processor groups. • Develop component interface for hybrid systems.	• Develop simple MCMD programming model. • Prototype hybrid interface for example application.	• Incorporate MCMD support for heterogeneous prog. models. • Implement hybrid & MCMD example application components.
Software Quality and Verification *Coordinator:* T.L. Dahlgren, LLNL; *Participants:* LLNL, ORNL *Motivating Applications:* fusion simulations, CQoS initiative			
• Identify and define CQoS and domain-specific semantics; assess spec. mechanisms.	• Develop semantics prototype(s). • Design method invocation sequencing constraints enforcement.	• Introduce semantic specifications into selected Toolkit components. • Develop sequencing enforcement prototype in Babel/SIDL.	• Evaluate semantics prototype(s). • Evaluate sequencing enforcement prototype. • Revise and evaluate prototypes based on CQoS evolution.
Computational Quality of Service (CQoS) *Coordinator:* L.C. McInnes, ANL; *Participants:* ANL, SNL *Motivating Applications:* combustion, quantum chemistry, accelerator, and fusion simulations			
• Collect application requirements, define metrics, perform base experiments. • Build database component.	• Populate CQoS testbed and specify initial CQoS API. • Develop initial performance models for applications. • Develop proxy port generation for CQoS usage.	• Complete design of overall CQoS strategy. • Implement application control laws. • Implement an asynchronous control infrastructure.	• Design APIs for general analysis engines. • Create a generic CQoS framework for HPC applications. • Stress test CQoS tools.

Table 4: **Summary of Milestones for CCA Environment**
Coordinator: G. Kumfert, LLNL; *Participants:* ANL, BU, LANL, LLNL, ORNL, SNL, UU

Year 1	*Year 2*	*Year 3*	*Years 4–5*
Core Tool Support and Maintenance *Coordinator:* B. Allan, SNL; *Participants:* ANL, LLNL, ORNL, SNL			
⟵ Support helpdesk and open bugtracking. ⟶			
⟵ Develop and maintain technical documentation. ⟶			
• Port CCA software stack to NLCF machines	• Complete CCA Conformance Tests	• Automated conformance testing for all CCA frameworks.	• Evaluate and port to new architectures as they emerge.
Enhancements *Coordinator:* T. Epperly, LLNL; *Participants:* BU, LLNL, ORNL, SNL, UU			
• Adopt EventService and MPIService into standard. • Demonstrate support for BabelRMI in XCAT.	• Demonstrate CCA/Kepler interoperability. • Add structs to SIDL/Babel. • Add Fortran 2003 support to Babel. • Incorporate SOAP as module in BabelRMI, integrate with Proteus/ XCAT.	• Finalize specification for Component sub-assemblies. • Develop specification for framework interoperability. • Release full fledged version of XCAT.	• Demonstrate exchange of sub-assemblies between two CCA Frameworks. • Demonstrate framework interoperability between CCA implementations. • Extend BabelRMI communication modules for new CCA applications.
Usability *Coordinator:* C. Rasmussen, LANL; *Participants:* LANL, LLNL, ORNL, SNL			
• Draft CCA-Lite Spec and CCA-Lite Framework. • Document advanced component debugging techniques. • Design component test harness.	• Preliminary integration of CCA-Lite test framework with Ccaffeine framework. • Deploy component test harness.	• Demonstrate connecting CCA-Lite components to CCA components in Ccaffeine. • Integrate SIDL semantics enforcement into testing methodology. • Develop component tracing tools to facilitate debugging.	• Demonstrate source-to-source conversion of CCA-Lite component to full CCA Component. • Evaluate tradeoffs in debugging and testing CCA-Lite vs. full CCA. • Apply test harness to selected toolkit components.

Table 5: **Summary of Milestones for the CCA Toolkit**
Coordinator: R. Armstrong, SNL; *Participants:* ANL, IU, LLNL, ORNL, PNNL, SNL, Tech-X, UMD, UU

Year 1	*Year 2*	*Year 3*	*Years 4–5*
⟵ Design, establish and, based on user feedback, iterate and improve CCA Base Installation ⟶			
• Design toolkit structure and contribute initial components to the Toolkit, and establish web distribution system	• Incorporate and promulgate Toolkit components into CCA tutorial and outreach activities, improve type and quality of the Toolkit repertoire.	• Add to Toolkit component improvements to CCA architecture since Year 1, e.g. MCMD components, templates, and CQoS plug-ins.	• Establish web-based/community process for approving/distributing component contributions from the community, as user base for Toolkit expands.

Table 6: **Summary of Milestones for Application and User Outreach and Support**
Coordinator: D.E. Bernholdt, ORNL; *Participants:* ANL, IU, LLNL, ORNL, PNNL, SNL, Tech-X

Year 1	*Year 2*	*Year 3*	*Years 4–5*
⟵ Support applications in adopting and using CCA. (See Appendix A entries labeled *Outreach* for likely partners.) ⟶			
	⟵ Deliver user support, incl. tutorials, coding camps, etc. ⟶		
	⟵ Update tutorial and best practices documentation. ⟶		
• Revamp `cca-forum.org` web services.	• Revamp or migrate `cca-forum.org` code development services.		

10 Institutional Roles and Key Personnel

10.1 Argonne National Laboratory

Argonne's work within the TASCS project will focus on several complementary areas. We will (1) develop computational quality of service (CQoS) infrastructure to help manage the composition, substitution, and reconfiguration of components in selected DOE scientific simulations, where possibilities include high-energy accelerators, fusion, quantum chemistry, and combustion; (2) design and implement components for parallel coupling based on the Model Coupling Toolkit, as motivated by simulations such as coupled core-edge fusion; (3) build numerical components for the CCA Toolkit, with emphasis on linear solvers, nonlinear solvers, and numerical optimization in collaboration with the TOPS project, as motivated by applications in fusion, accelerator modeling, and quantum chemistry; (4) develop tools to facilitate the usability of CCA components, including Eclipse IDE capabilities as well as fully automated build and configuration support; and (5) write tutorial materials and example component codes to teach CCA concepts to the high-performance computing community.

Personnel

The list of senior personnel at Argonne National Laboratory participating in this project is as follows:

- **Lois Curfman McInnes** coordinates the Component Technology Initiatives focus of the project and is the institutional lead PI for Argonne National Laboratory. She has the responsibility for participating in the management team, including oversight, management, and reporting of activities at Argonne. Her technical involvement will be primarily in coordinating the Computational Quality of Service Initiative and in developing components for linear solvers, nonlinear solvers, and numerical optimization as part of the CCA Toolkit. She will also participate in Application Outreach.

- **Boyana Norris** will be involved in most technical aspects of the project, with particular emphasis on designing and building CQoS infrastructure, developing components for linear and nonlinear solvers within the CCA Toolkit, and enhancing CCA usability by developing Eclipse IDE capabilities and fully automated build and configuration support. She will also participate in Application Outreach.

- **Jay Larson** will focus on developing components for parallel coupling as part of the CCA Toolkit; these capabilities are motivated by simulations such as core-edge coupled fusion.

10.2 Binghamton University

The overall objective Binghamton's involvement is to work on enhancements of the CCA environment, motivated by the applications and research initiatives discussed in the proposal. Binghamton University will add powerful new capabilities, including support for BabelRMI in the distributed XCAT framework, design and development of new communication providers for BabelRMI including Web services based protocols to facilitate interaction with workflow tools and applications, and development of mechanisms to enable seamless interoperability between CCA framework and applications.

Personnel

The list of senior personnel participating in this project at Binghamton University is as follows:

- **Madhusudhan Govindaraju** is the institutional lead PI for Binghamton University. He has the responsibility for participating in the management team, oversight, and reporting for activities at Binghamton University. He will be involved with the technical aspects of the proposal with particular focus on the integration of BabelRMI with XCAT, distributed CCA frameworks, and minor participation in other areas.

- **Kenneth Chiu** will be involved in the technical aspects of the proposal with emphasis on design and development of new modules for BabelRMI, design of a multi-protocol communication substrate for XCAT, and distributed framework interoperability.

- **Michael J. Lewis** will be involved in the technical aspects of the proposal with emphasis on design and development of a thorough specification and implementation modules to enable interoperability between different CCA frameworks.

10.3 Indiana University

IU's primary responsibility in TASCS is in the toolkit initiative, in particular components for a shared linear solver interface, parallel I/O, and data management. These will be developed and tested initially in the context of code coupling.

Part of the toolkit effort is installation of software and maintenance on the cluster reserved for CCA use and will be an important testbed for the toolkit, outreach, and base installation efforts. This extends across all years, as does systems support for TASCS tutorials and coding camps on the cluster.

Personnel

The list of senior personnel participating in this project at Indiana University is as follows:

- **Randy Bramley** has responsibility for managing and reporting IU's activities, and will have primary responsibility for maintaining hardware and systems software support for outreach and tutorial activities, and the development of components for parallel I/O and data managment. He is also a primary point of contact for fusion energy simulation applications as a participant in the Center for Simulation of Wave Interactions with Magnetohydrodynamics (SWIM) and the newly-funded Framework Application for Core-Edge Transport Simulations (FACETS) SciDAC projects.

10.4 Lawrence Livermore National Laboratory

LLNL is the lead institution for the CCA Environment and birthplace of SIDL/Babel. It is also the lead for the Software Quality and Verification Initiative, which is focused on automating the use of domain-specific behavioral and Computational Quality-of-Service (CQoS) semantics for scientific software components. In addition, LLNL participates in most of the Center's other activities — from research initiatives to user outreach.

In the Environment, we are heavily involved in supporting core tools, and broadening our established software engineering discipline (testing, porting, customer support) to other CCA products. We are committed to a series of technical enhancements to SIDL/Babel as part of our support to SciDAC customers as well as internal Initiatives.

Historically, we have also been well represented in the usability and outreach efforts of the CCA. In addition, we expect to perform ongoing support services to users, participate in tutorials, coding camps, etc.

Personnel

The list of senior personnel at Lawrence Livermore National Laboratory participating in this project is as follows:

- **Gary Kumfert** coordinates the CCA Environment, is institutional lead PI for LLNL, and is co-architect of Babel (with Tom Epperly). Coordinating the largest of the four main thrust areas, Gary orchestrates the efforts of seven institutions in the basic maintenance, technical improvements, and usability improvements of CCA technology. As LLNL-PI he is responsible for oversight, management, and reporting activities of his institution. His primary technical responsibility is shepherding the CCA specification through the standardization process to 1.0.
- **Tammy Dahlgren** coordinates the Software Quality and Verification initiative and participates in the Computational Quality-of-Service (CQoS) effort. Her technical responsibilities are focused on the integration of Design-by-Contract enforcement and CQoS semantics support mechanisms into SIDL/Babel on behalf of the CCA.
- **Tom Epperly** coordinates the Enhancements intiative for the CCA Environment area. Along with Gary Kumfert, he is the co-architect of Babel. Tom will be working on adding structs and Fortran 2003 support to Babel, and he also takes an active role in supporting users. and addressing bug reports. Tom is the liason with the FACETS project.

10.5 Los Alamos National Laboratory

LANL will lead the Usability activity, within the CCA Environment focus area, and work primarily on “CCA-Lite” development.

Personnel

The list of senior personnel participating in this project at Los Alamos National Laboratory is as follows:

- **Craig Rasmussen** is the LANL institutional lead PI, and as such has responsibility for oversight, management, and reporting of activities at LANL. His technical involvement will be primarily in the CCA-Lite effort, but will also play a minor role in the Environment Usability Activity.
- **Christopher Rickett** will help with the CCA-Lite specification effort and will be responsible for its implementation.

10.6 Oak Ridge National Laboratory

ORNL is the overall lead institution for TASCS and has some degree of involvement in most of the Center’s technical activities. We will participate in the Emerging HPC Paradigms and Software Quality Component Technology Initiatives. We will participate in the CCA Environment and CCA Toolkit Focus Areas, including exploring technology and interfaces for parallel coupling. Finally, ORNL leads the Application and User Outreach and Support Focus Area.

In the Emerging HPC initiative, we will be involved in the MCMD/process group design and implementation, and in fault tolerance activities. In the Software Quality initiative, we will focus on integrating semantic checking with the component unit testing framework.

In the CCA Environment focus area, we will play a secondary role in the maintenance, support and porting of the core tools, productizing the component sub-assembly tools, and developing a unit testing harness for CCA components and other debugging tools. In the CCA Toolkit, we will be actively involved in work on the “CCA base installation” and the development of a variety of components.

ORNL leads the Application and User Outreach and Support focus area, and will actively participate in aiding outside projects working with the CCA, refining and presenting the tutorial and related documentation, and a broad range of community outreach activities.

Personnel

The list of senior personnel participating in this project at Oak Ridge National Laboratory is as follows:

- **David Bernholdt** is the Lead PI of the project. He has overall responsibility for oversight, management, and reporting for the project. His technical involvement will be primarily in the User and Application Outreach and Support area, with some participation in the Software Quality Initiative, the Environment Usability activity, and the CCA Toolkit.
- **Wael Elwasif** will be involved in most technical aspects of the project, focusing particularly on the Software Quality Initiative and the Environment Usability Activity.
- **James Kohl** is the ORNL institutional lead PI, and as such has responsibility for participating in the management team, including oversight, management, and reporting of activities at ORNL. His technical involvement will focus on components for parallel coupling as part of the CCA Toolkit and the Emerging High-Performance Software Paradigms Initiative, with minor participation in other areas.

10.7 Pacific Northwest National Laboratory

The main focus of PNNL work will be on Emerging HPC Initiative. In addition, the PNNL team will be involved in advancing the CCA Toolkit in terms of developing HPC components and improving its usability. PNNL will be engaged in outreach activities with special focus on biology, chemistry, and groundwater applications.

Personnel

The list of senior personnel at Pacific Northwest National Laboratory participating in this project is as follows:

- **Jarek Nieplocha** is a co-PI for the project and lead PI for the PNNL work. He will also lead the Emerging HPC initiative for the project. He has overall responsibility for oversight, management, and reporting for the project activities at PNNL. His technical involvement will be in the area of multiple-component-multiple-data model and component technology for hybrid platforms.
- **Manojkumar Krishnan** will be in the following areas: multiple-component-multiple-data, development of HPC components of the CCA toolkit and improving its usability. He will be also also in application outreach activities.
- **Ian Gorton** will be involved in technical activities under the Emerging HPC initiative and outreach. One the specific activities will be development of the event model as needed to support heterogeneous platforms.
- **Daniel Chavarria** will be working on component technology in support of heterogenous platforms and biology application outreach.

10.8 Sandia National Laboratories

Sandia has responsibility for the HPC framework Ccaffeine. While all of the participants in the center will be involved with modifications to the CCA specification to accomodate computational quality of service (CQoS), MCMD, and other activities, Sandia must implement these in Ccaffeine, a formidable task. Sandia also will lead work on the CCA Toolkit and will spearhead the "product" of this center: the CCA base installation and components for the toolkit. Sandia will contribute a pivotal data model component, the Structured Mesh Component. Sandia will also contribute general-purpose utility components (e.g., I/O, MPI, etc.) to the toolkit. Sandia is also active in creating software for CQoS experimentation and design.

Personnel

The list of senior personnel at Sandia National Laboratories participating in this project is as follows:

- **Rob Armstrong** is the lead PI for the Sandia effort and current Chair of the CCA. He is coordinating the CCA Toolkit effort and contributes to the CCA Lite design and implementation.
- **Ben Allan** is the principal author of Ccaffeine, the most commonly used CCA framework. He is a contributor to the CCA Lite design and implementation and helps implement the tutorial source code and toolkit tools.
- **Jaideep Ray** is a contributor to the Computational Quality of Service work for combustion and other CCA applications.

10.9 Tech-X Corporation

The overall objective of Tech-X involvment is to work the CCA Base Installation (Toolkit), and the Application Support focus areas (Outreach). More specifically, Tech-X will participate in the process of Components Debugging and Testing and provide "walk-up" connections with the domain specific applications (fusion).

Personnel

The list of senior personnel participating in this project at Tech-X Corporation is as follows:

- **Svetlana Shasharina** is the institutional lead PI for Tech-X Corporation. She is responsible for managing all activities and personnel at Tech-X. She will participate in the Toolkit (installation) and Outreach (fusion) activities.
- **Johan Carlsson** is a physicist and will work on Outreach (developing fusion-specific components and providing "walk-up" connections with fusion applications).
- **Nanbor Wang** is a computer scientist and will work on Toolkit (components testing and debugging).

10.10 University of Maryland

UMD's primary responsibility in TASCS is in the toolkit, especially work associated with parallel I/O and coupling entire existing codes and the data and metadata management that requires. One key part of the project is integrating some of the capabilities of the Maryland InterComm framework as components in the toolkit, as initial work towards code coupling. The other part of the project is integrating spatial indexing techniques and libraries already developed at UMD into the parallel I/O interfaces and implementations.

Personnel

The list of senior personnel participating in this project at the University of Maryland is as follows:

- **Alan Sussman** is the institutional lead PI for University of Maryland. He is responsible for managing all activities and personnel at Maryland, including a research programmer and graduate student. He will participate in and manage the development at Maryland of the parallel I/O and code coupling components for the CCA Toolkit, working with TASCS collaborators at other sites.

- **Norman Lo** is a research programmer, who will work on the design and implementation of CCA Toolkit components for parallel I/O and code coupling.

10.11 University of Utah

Utah is responsible for technology in integrating CCA with other workflow-based, component-based and object-based programming paradigms, such as Kepler from the Scientific Data Management Center. Utah will also contribute simple CCA interfaces to existing visualization software, and will work with application teams to deploy these tools in the CCA toolkit. Finally, Utah will be involved with developing tools and standards to ensure interoperability between the disparate CCA frameworks.

Although not a direct focus of this proposal, Utah will also continue to develop and maintain the SCIRun2 CCA framework.

Personnel

The list of senior personnel participating in this project at the University of Utah is as follows:

- **Dr. Steven Parker** will manage the effort at Utah, overseeing the progress of students and staff that contribute to the above-mentioned goals. Dr. Parker will also help develop specifications for standard interfaces, and will partipate with TASCS collaborators to enable interoperability between TASCS software and other external packages.

11 Project Budget

The planned budget for the project is shown below.

	Budget ($k)						
Institution	FY06	FY07	FY08	FY09	FY10	FY11	Total
Argonne National Laboratory	168	505	505	505	505	337	2525
Los Alamos National Laboratory	31	94	94	94	94	63	470
Lawrence Livermore National Laboratory	179	537	537	537	537	358	2685
Oak Ridge National Laboratory	167	500	500	500	500	300	2500
Pacific Northwest National Laboratory	179	329	329	329	329	219	1645
Sandia National Laboratories	179	538	538	538	538	359	2690
Lab Total	834	2503	2503	2503	2503	1669	12515
Binghamton University	0	93	93	93	93	93	465
Indiana University	0	127	127	127	127	127	635
Tech-X Corporation	0	87	87	87	87	87	435
University of Maryland	0	62	62	62	62	62	310
University of Utah	0	128	128	128	128	128	640
University Total	0	497	497	497	497	497	2485
Center Total	834	3000	3000	3000	3000	2166	15000

A Prospective Collaborations

The following table lists projects we anticipate collaborating with. This list is based primarily on the list of potential collaborations in our proposal which either have been funded, or are still under consideration, but includes additional prospective partners we've encountered more recently.

The list is divided into groups based on the "reality" of the collaboration. Some collaborations are already *active* (especially on-going interactions, carried over from CCTTSS). In other cases, the interactions are *developing* (especially for partners which are newly funded, but with whom we've had extensive preliminary discussions). Finally, there are cases where we have a mutual interest in collaboration, but we are just beginning to *explore* the possibilities. We will provide updated information as to the status of our collaborations through our regular progress reports and other mechanisms, as requested.

The table briefly described the nature of the interaction from the TASCS viewpoint, and indicates the point of contact on the TASCS side who is the primary liaison with each project. Frequently, the point of contact and other TASCS participants are also direct (funded) participants in the partner project.

The final column of the table indicates which element of the project the collaboration is associated with, according to the breakdown given in the Technical Overview (Section 2). As described in Section 4, we have different mechanisms for interacting with our partners, which are also reflected in our Milestones and Deliverables (Section 9). Partnerships associated with the *Initiatives* and *Environment* portions of TASCS typically provide direct motivation for TASCS research activities. Those listed under *Toolkit* are expected to result in common interfaces and reusable components. Finally, those listed under *Outreach* are expected to be general users of CCA technology who will be supported as part of our Application Outreach activities.

Domain: Collaborating Project (PI)	**Project Type**	**TASCS Activities (Point of Contact)**	**TASCS Project Element**
		Active or On-Going Collaborations	
Accelerators: A Beam Dynamics Application Based on the CCA (Douglas Dechow, Tech-X)	OASCR SBIR	Developing a component-based beam dynamics application from a set of services that are provided by 3 separate applications (Lois McInnes, ANL)	Outreach
Applied Math: Center for Interoperable Technologies for Advanced Petascale Simulations (ITAPS) (Lori Diachin, LLNL)	SciDAC-2 CET	Defining common interfaces for scalable mesh and discretization technologies (Tammy Dahlgren, LLNL)	Toolkit
Applied Math: Towards Optimal Petascale Simulations (TOPS) (David Keyes, Columbia U.)	SciDAC-2 CET	Developing common interfaces and components for scalable linear solvers, nonlinear solvers, and numerical optimization (Lois McInnes, ANL)	Toolkit
Chemistry: Chemistry Framework Using the CCA (Mark Gordon, Ames Lab)	SciDAC-2 SAP	Developing component infrastructure for quantum chemistry; building tools for adaptively managing MCMD simulations (Joe Kenny, SNL)	Initiatives/HPC & CQoS

(continued)

(continued)

Domain: Collaborating Project (PI)	**Project Type**	**TASCS Activities (Point of Contact)**	**TASCS Project Element**
Combustion: Computational Facility for Reacting Flow Science (Habib Najm, SNL)	SciDAC-2 App.	Developing a component-based combustion toolkit; building tools to adaptively manage parallel partitioning strategies (Jaideep Ray, SNL)	Initiatives/CQoS
Computer Science: Collaborative Research: Interactive Parallel Platforms for Multi-Experiment Computational Studies (Denis Zorin, NYU)	NSF	Developing framework support for parameter study applications (Steve Parker, UU)	Outreach
Computer Science: Distributed CCA Components and Grid Services for Scientific Computing (Nanbor Wang, Tech-X)	OASCR SBIR	Developing remote capabilities for CCA applications (Nanbor Wang, Tech-X)	Outreach
Computer Science: Scientific Data Management Center for Enabling Technologies (Arie Shoshani, LBNL)	SciDAC-2 CET	Developing interoperability between the Kepler workflow system and the CCA (Steve Parker, UU)	Env./Enhancements, Toolkit
Fusion: Framework for Modernization and Componentization of Fusion Modules (FMCFM) (Johan Carlsson, Tech-X)	OFES SBIR	Modernize modules extracted from established fusion codes with subsequent conversion of these modules to components (Johan Carlsson, Tech-X)	Outreach
Fusion: Simulation of Wave Interactions with Magnetohydrodynamics (SWIM) (Don Batchelor, ORNL)	SciDAC-1 App.	Using CCA technologies to create new fusion simulation capabilities by coupling together existing fusion applications (Randall Bramley, IU)	Outreach
Materials Science and Computer Science: Petascale Simulation Initiative (John May, LLNL)	LLNL LDRD	Developing Babel-based parallel RMI capabilities for petascale machines, to be used by MPMD multiscale materials simulations (Gary Kumfert, LLNL)	Initiatives/HPC, Env./Enhancements
Nanotechnology: Computational Nanophotonics: Modeling Optical Interactions and Transport in Tailored Nanosystem Architectures (Stephen Gray, ANL)	OBES App.	Leveraging component technology to provide a suite of nanophotonic simulation software for modeling light interacting with nanoparticles (Boyana Norris, ANL)	Outreach

(continued)

(continued)

Domain: Collaborating Project (PI)	**Project Type**	**TASCS Activities (Point of Contact)**	**TASCS Project Element**
Nuclear Energy: Terascale Simulation Tools for Next-Generation Nuclear Energy Systems (Kevin Clarno, ORNL)	ORNL LDRD	CCA-based componentization to integrate models for advanced reactors (Wael Elwasif, ORNL)	Outreach
Performance: Knowledge-based Parallel Performance Technology for Scientific Application Competitiveness (Allen Malony, U. Oregon)	OASCR Base (in review)	Providing CCA interfaces to performance tools; providing performance data and analysis infrastructure for CQoS initiative (Jaideep Ray, SNL)	Initiatives/CQoS, Toolkit
	Developing Collaborations		
Applied Math: The Applied Partial Differential Equations Center (APDEC) (Phil Colella, LBNL)	SciDAC-2 CET	Defining common interfaces and components for adaptive mesh refinement (Jaideep Ray, SNL)	Toolkit
Biology: Data Intensive Computing for Complex Biological Systems (Tjerk Straatsma, PNNL)	OASCR Base	Deploying CCA for the Polygraph proteomics application, which uses heterogeneous architectures (Jarek Nieplocha, PNNL)	Initiatives/HPC
Climate: A Data Domain to Model Conversion Package for Sparse Climate Related Process Measurements (Rao Kotamarthi, ANL)	SciDAC-2 SAP	Leveraging component technology to develop a uniform set of software tools suitable for the evaluation of high-end climate models (Jay Larson, ANL)	Outreach
Computer Science: Coordinated Fault Tolerance (Pete Beckman, ANL)	OASCR Base	Making fault tolerance backplane capabilities accessible to the CCA community (David Bernholdt, ORNL)	Initiatives/HPC, Toolkit
Fusion: Framework Application for Core-Edge Transport Simulations (FACETS) (John Cary, Tech-X)	SciDAC-2 App.	Developing a component-based framework for core-edge fusion simulations (Tom Epperly, LLNL)	Env./Enhancements, Toolkit, Initiatives/CQoS
Groundwater: Hybrid Numerical Methods for Multiscale Simulations of Subsurface Biogeochemical Processes (Timothy Scheibe, PNNL); *Embedded SAP:* Component Software Infrastructure for Achieving High Level Scalability in Groundwater Reactive Transport Modeling and Simulation (Bruce Palmer, PNNL)	SciDAC-2 App./SAP	Building component infrastructure for subsurface simulations (Manoj Krishnan, PNNL)	Outreach

(continued)

(continued)

Domain: Collaborating Project (PI)	**Project Type**	**TASCS Activities (Point of Contact)**	**TASCS Project Element**
Space Physics: Center for Integrated Space Weather Modeling (Jeffrey Hughes, Boston U.)	NSF	Developing coupled simulations of the environment between the sun and earth; motivating development of parallel component coupling tools (Alan Sussman, UMD)	Outreach
Visualization: Visualization and Analytics Center for Enabling Technologies (Wes Bethel, LBNL)	SciDAC-2 CET	Providing CCA-based interfaces for visualization tools (Steve Parker, UU)	Toolkit
Exploring Collaboration Opportunities			
Applied Math: Combinatorial Scientific Computing and Petascale Simulations (Alex Pothen, Old Dominion U.)	SciDAC-2 Inst.	Leveraging TASCS technology in the definition of component interfaces and implementations for graph algorithms (Boyana Norris, ANL)	Toolkit
Computer Science: Center for Scalable Application Development Software (Ken Kennedy, Rice U.)	SciDAC-2 CET	To be determined (David Bernholdt, ORNL)	Outreach
Performance: Performance Engineering Research Institute (PERI) (Robert Lucas, USC)	SciDAC-2 Inst.	Leveraging performance evaluation tools for component simulations and CQoS initiative (Boyana Norris, ANL)	Initiatives/CQoS
Stellar Dynamics: Collaborative Research: An Integrated Software Environment for Multiscale Studies of Galactic Nuclei and Compact Stellar Systems (Stephen McMillan, Drexel U.)	NSF (in review)	Coupling several different stellar models; motivating development of parallel coupling tools (David Bernholdt, ORNL)	Outreach
Stellar Dynamics: Collaborative Research: Simulation of Dense Stellar Systems on Special-Purpose Supercomputers (David Merritt, Rochester Inst. Tech.)	NSF (in review)	Coupling several different stellar models; motivating development of parallel coupling tools (David Bernholdt, ORNL)	Outreach