Software Design and Productivity

Definition of SDP PCA

The activities funded under the Software Design and Productivity (SDP) PCA will lead to fundamental advances in concepts, methods, techniques, and tools for software design, development, and maintenance that can address the widening gap between society's need for usable and dependable software-based systems and the ability to produce them in a timely, predictable, and cost-effective manner. The SDP R&D agenda spans both the engineering components of software creation and the economics of software management across all IT domains, including the emerging areas of embedded systems, sensor networks, autonomous software, and highly complex, interconnected systems of systems. Today, software development and maintenance represent by far the most costly, time-consuming, labor-intensive, and frustrating aspects of IT deployment in every economic sector. SDP R&D seeks to foster a new era of innovation in software engineering that addresses these serious design issues.

Broad Areas of SDP Concern

- Overall quality of software
- Overall cost – in time, labor, and money – of software development and maintenance
- Growing complexity of software
- Enabling more people to more easily create software and software-based systems
- Need for expertise in emerging software areas such as embedded systems, large-scale sensor networks, autonomous software, and grid environments
- Workforce development

Technical Goals

- Scientific foundations for creating, maintaining, and improving software that incorporates such qualities as usability, reliability, scalability, and interoperability
- More cost-effective methods for software testing, analysis, and evaluation
- New frameworks for understanding and managing the economics of software
- Methods for developing and evaluating specialized software systems for specific purposes and domains

Illustrative Technical Thrusts

- Development methodologies such as model frameworks; tunable, adaptable processes; component technologies; open source practices for code portability and re-use; integrated environments and tools for development; and programming standards
- Theoretical and technical aspects of programming languages; compilers; software for visualizing and verifying code and data structures; and automatic program synthesis
- Software architectures and component-based methods to incorporate re-usable software resources
- Measuring and managing software quality
- Scalability: Enhancing the ability of software to "grow" along various axes without significant redevelopment
- Interoperability: Software that moves easily across heterogeneous platforms as well as software units that cooperate and exchange information seamlessly
- Fault-tolerant, adaptable software that can continue to function under unexpected conditions

SDP Agencies

			Participating Agency
NASA	DOE/NNSA	NIH	
NSF	NIST	NOAA	FAA

SDP PCA Budget Crosscut

FY 2004 Estimate	FY 2005 Request
$179.3 M	$166.0 M

- Enhancing software performance through code modeling and measurement; understanding how to improve runtime effectiveness of applications
- Techniques and tools - including automated approaches, new metrics, and reference data - for testing, analysis, validation, and verification
- Techniques and tools for assessing engineering risks and tradeoffs and estimating and balancing developmental and operational costs
- Specialized and domain-specific software such as for application frameworks, platform reliability, and modeling or control of embedded systems

SDP PCA: Coordination and Activities

SDP Highlights

Since the inception of the SDP PCA in FY 2000, its Coordinating Group (CG) has worked on developing a common understanding of the SDP R&D terrain. In this process, it has worked with the HCSS CG to make clear the research distinctions between their two domains (see HCSS definition on page 64). These efforts contribute to a broader understanding by the policymaking community and the public of software issues and the critical need for R&D that addresses them.

In December 2001 at the outset of its deliberations, the SDP CG sponsored a workshop at which researchers from academe, industry, and government were encouraged to "think out of the box" about "New Visions for Software Design and Productivity." The participants identified five major facets of the growing problems in contemporary software design and development:

- The large number and inherent complexity of requirements
- Multiple independent sets of requirements and their associated functionalities
- Testing, verification, and certification of large, complex systems
- Composition and integration of software from multiple sources
- Multidisciplinary distributed software development

Building on that workshop's conclusions, in FY 2004 the SDP CG will complete a taxonomy of the PCA's scope of research. The preliminary taxonomy appears

Continued on next page

Scope of SDP R&D Topics

Create, maintain, and improve software

- Software development methods
  - Model development framework
  - Tunable development processes
  - Component technologies
  - Open source
  - Development environments
  - Programming standards
  - Adaptable systems
  - Legacy systems
- Languages and tools
  - Programming languages
  - Cross-platform Fortran 90
  - Compilers
  - Programming environments
  - Software visualization
  - Automatic program synthesis
- Runtime environment

Investigating and achieving the "-ilities"

- Quality management
- Platform reliability
- Scalability
- Interoperability
- Robustness
- Performance tuning

Testing, analysis, and evaluation

- Evaluation of complex integrated systems with real-time characteristics
- Evaluation of models and simulations
- Evaluation of COTS
- Metrics and reference data
- Automatic verification and validation (V&V)
- Testing tools
- Software analysis

Management and economics

- Tools for various aspects of software process management
  - Project management
  - Cost and schedule estimation and prediction
  - Cost of testing
  - Document management systems
  - Quality vs. risk measurement and management

Software systems for specific domains and specialized software

- Application frameworks
  - Domain-specific tools etc.
  - Specialized software products
  - Large-scale data and information access, sharing, security, etc.
  - Scientific data management
  - Visualization and analysis of scientific data sets
- Software for platform reliability
  - Software to model platform architecture
- Embedded systems
  - Software control
  - Modeling and simulation
  - Self-evolving (adaptable) systems

Briefings with IT 'User' Agencies

In FY 2004, the SDP CG is holding briefings with representatives of Federal "IT user" agencies to gather information about software issues from agencies whose missions involve large-scale, critical applications – such as for Social Security records, Medicare transactions, and Internal Revenue systems. The goal is to better understand what IT managers – from their firsthand practical knowledge – perceive to be the key problems of their complex, real-time software environments. The first briefing of the series was provided by the DHS Customs and Border Protection agency, describing development of a new distributed IT system to manage U.S. export-import procedures.

Multiscale Modeling Workshop Planned

The SDP CG also is planning an FY 2005 workshop to explore software issues in multiscale modeling and simulation of complex physical systems. Modeling and simulation software has become a principal research tool across the sciences, and creating models that combine multiple factors at varying scales – such as climate from global to local scales, or interacting systems in the human body through the shape and functions of protein molecules – is both a key opportunity for scientific discovery and a key challenge for high-end computing and domain sciences. The workshop will bring computer and domain scientists together to assess the state of the art and possible synergies among computational tool sets developed for differing scientific domains.

Multiagency SDP Activities

The following is a sampling of FY 2004 activities in which more than one NITRD agency participates (other collaborating agencies are not cited):

NASA – Software design for Earth System Modeling Framework (ESMF); partners include DOE/SC, NOAA, and NSF; problem-solving frameworks, with DOE/SC and NSF; automated software engineering, with NSF; joint work on grid software with DOE/SC and NSF through the Global Grid Forum. (For more details on ESMF, see page 13.)

NSF – collaborates with NASA, DARPA, and others with overlapping missions, such as in the Highly Dependable Computing and Communications Systems Research (HDCCSR) program, with NASA, and ESCHER, with DARPA (see below)

DOE/NNSA – the Advanced Simulation and Computing (ASC) program is open to collaborations and dialogues with other agencies, such as funding R&D in open source software for high-performance computing. ASC researchers collaborate with ASC university alliance members that are also funded by other NITRD agencies, strengthening the collaborative environment in academic research and education.

DARPA – the Model-Based Integration of Embedded Systems (MoBIES) program has an interagency activities aspect, the Embedded Systems Consortium for Hybrid & Embedded Research (ESCHER), with NSF. ESCHER is a repository of technologies and tools from DARPA and NSF programs in hybrid and embedded systems. The prototypes and documentation are intended to promote rapid transition of MoBIES results to DoD and industry.

NIH – holding preliminary discussions with other agencies about an FY 2005 multiagency workshop on informatics associated with microbial science

NIST – collaborates with a wide range of Federal organizations on SDP-related topics, including:

- Data uniformity and Standards Structural Bioinformatics (includes Protein Databank, HIV Structural Database), with DOE/SC, NIH, NSF, and universities
- Digital Library of Mathematical Functions, with NSF
- NeXus Data Exchange Standard (effort to get neutron researchers to use sharable data structures), with DOE/SC
- Numerical Data Markup Language (developing an XML schema of encoding units of measurement, called UnitsML), with DOE/SC
- Product Data Standards for HVAC/R, with DOE
- Product Engineering Program (role of CAD systems; metrics for interoperable representation standards in product engineering), with DARPA, NASA

SDP R&D PROGRAMS BY AGENCY

SELECTED FY 2004 ACTIVITIES AND FY 2005 PLANS

SDP NASA SDP

NASA missions are increasingly and critically dependent on complex software systems; in many cases, software cannot be repaired or reloaded after a mission begins. The agency therefore has a critical need for:

- Automated software verification and validation tools
- Automated program synthesis
- Evolvable and adaptable software

Science and engineering applications in general are becoming more complex and multidisciplinary, and require distributed computing including "high-performance" applications, integration and interoperability of independent components, and runtime support for efficient discovery, access, and use of distributed resources including data and software. The agency uses a Technology Readiness Level (TRL) scale that runs from 1 (basic concept) to 9 (embedded in a mission) to categorize R&D.

FY 2004 agencywide SDP R&D activities are located in the Computing, Information, and Communications Technology (CICT) Program, Engineering for Complex Systems (ECS) Program, and Earth Science Technology Office (ESTO), in the following programs:

Information Technology Strategic Research, CICT

Automated Software Engineering Technologies (with NSF) – automated mathematical techniques for the software development process, yielding tools for cost-effective development of high-confidence, highly reliable software systems for aerospace applications. Low TRL. Technology development includes:

- Scalable software model checking, automated program abstraction, state-space search algorithms, and formal method verification of integrated modular avionics design to make possible analytical verification of concurrent advanced aerospace software architectures and code
- Program generation through automated reasoning, product-oriented certification methods, and automated tools that certify automatically synthesized code
- Generate runtime monitors from requirements specifications, automated behavioral verification, machine learning to optimize exploration of potential behaviors, and automated generation of software fault recovery, to enable software that monitors itself and recovers from faults at runtime with minimal computational overhead

Evolvable Systems – aiming to dramatically increase mission survivability and science return through development and application of evolutionary and adaptive algorithms. Includes:

- Advanced evolutionary algorithms and automated design and optimization for evolved fault recovery, to enable programmable logic that automatically rewires following damage
- Evolved sensor suite and evolutionary scheduling algorithms to enable sensor electronics that survives extreme radiation and temperature
- Future developments may include defect-tolerant nanosystems, distributed control, and evolved control algorithms for evolvable robotic control and coordination

Intelligent Systems, CICT

Verification and Validation (V&V) for Autonomy – addresses the complexity in verifying autonomous systems that operate in rich and uncertain environments with no human intervention, and that must adhere to internal correctness constraints involving communication among components, control flow, and resource utilization. This project to develop state-of-the-art benchmarking tools for V&V has successfully used rover software to test experimental versions of V&V tools. TRL 3.

Computing, Networking, and Information Systems, CICT

Problem Solving Frameworks (with DOE/SC, NSF) – objective is to develop infrastructures that support the management of simulation, analysis, and data components in application-specific environments. Challenges:

- Component representations that support the access, use, and composition of multi-component applications

- Methods for automatic and user-based generation of work plans
- Efficient management of the flow of work across distributed resources

Projects include:

– Component and Data Representation Models

– GridWorks: Workflow Management Framework

– CIAPP: CORBA-based Framework

– TAF-J: Agent-based Framework

– AeroDB

Grid Middleware Services (with DOE/SC, NSF) – developing a distributed infrastructure for seamless access to resources that are heterogeneous (computers, data, and instruments), distributed across multiple administrative domains, and dynamically coupled. This involves developing grid services (e.g., application and user-oriented, execution management, grid management, and data management and access) that provide a useful function independent of the underlying resources, are discoverable, encourage the design and use of reusable software components, and are easily incorporated into application frameworks. The runtime aspects of this program also fall under LSN's MAGIC effort.

Parallel Programming Paradigm – seeks to improve parallel programming tools and techniques to increase the performance of HEC systems scaling to thousands of processors, often in a grid environment of multiple distributed platforms. The portability, scalability, and usability of codes are technology challenges. The effort is evaluating (and, if necessary, extending standards for) parallel multilevel programming paradigms. Exploring: Multilevel Parallelism (MLP); MPI + OpenMP Hybrid Parallelism; and Distributed Shared Memory

Automated Programming Tools for Parallelization – aims to reduce time-consuming and error-prone code porting and development times for applications to run on HEC systems and grids by providing an integrated set of automated tools for parallelization that retains good performance. Challenges:

- Straightforward porting process
- Accurate, effective code analysis
- Multi-level parallel code generation
- Correct answers with ported code
- Performance analysis
- Good performance

Projects include:

– CAPO: Computer-Aided Parallelizer and Optimizer

– Performance analysis

– ADAPT: Automatic Data Alignment and Placement Tool

– Automatic debugging

Performance Modeling, Benchmarking, and Optimization – techniques, strategies, and tools to model, predict, and optimize application performance, and to improve applications' maintainability, portability, and performance in parallel and distributed heterogeneous environments. Technical approach:

- Predicting application execution times on various platforms
- Predicting wait times in scheduler queues
- Developing benchmarks to understand the behavior of key NASA applications on single (parallel) and distributed systems
- Understanding application cache usage and optimizing performance of regular and irregular problems

Projects are:

– NAS Parallel Benchmark

– NAS Unstructured Mesh Benchmarks

– NAS Grid Benchmarks

– Performance prediction

– Cache optimization

Resilient Software Engineering, ECS Program

High Dependability Computing – developing tools, case studies, and testbeds to identify and characterize risk precursors in mission software systems. NASA has a user collaboration portal for design and development; computer scientists and mission experts work together on dependability models and metrics. Testing is done on rover software and onboard spacecraft IT systems. The goal is to transfer successful dependability metrics and technologies to missions and industry. High TRL.

Intelligent Software Engineering Software Suite – investigating software algorithms, processes, and development procedures to improve software integrity and reliability. The work begins with the study of critical NASA software risks and prototype tool productization (methodology development, tool enhancement, and beta testing), then moves to a tool maturation and evaluation

stage (methodology integration and tool customization), and finally to integration in mission processes and adoption by mission partners. Low TRL.

Earth System Modeling Framework (ESMF), ESTO

NASA leads this collaboration of many agencies and academic researchers in building high-performance, flexible software infrastructure to increase ease of use, performance portability, interoperability, and reuse in climate, numerical weather prediction, data assimilation, and other Earth science applications. See story on page 13.

SDP NSF SDP

Applications software development is supported across CISE and specific applications (such as in biology, chemistry, and physics) are developed in other directorates. Across all NSF directorates, more emphasis is being placed on cyberinfrastructure, including software to support the science and engineering enterprise. In FY 2004, a new, cross-divisional "Science of Design" theme is being developed in CISE. SDP work is also funded under the ITR program, which involves all NSF directorates. As of December 2003, there were 1,300 active ITR awards, about 10 percent of which involve SDP research.

CISE SDP project areas include:

- Software Engineering and Languages (SEL)
- Advanced Scientific Computing (ASC)
- Embedded and Hybrid Systems (EHS)
- Next Generation Software (NGS)
- Distributed Systems and Compilers (DSC)

The current NSF-funded topics can be mapped into the SDP CG's taxonomy (page 55) as follows:

- Create, maintain, and improve software
  - Software development methods – ASC, EHS, NGS, SEL, ITR
  - Languages and tools – ASC, DSC, NGS, SEL, ITR
  - Runtime environments – DSC, NGS, ITR
- Investigating and achieving the "-ilities"
  - Platform reliability – DSC, ITR
  - Robustness – EHS, SEL
- Testing, analysis, and evaluation
  - Evaluation of models and simulations – ASC, DSC, NGS, ITR
  - Metrics and reference data – SEL, ITR
  - Automatic V&V – SEL, ITR
  - Testing tools – SEL, ITR
  - Software analysis – SEL, ITR
- Management and economics – SEL, ITR
- Software systems for specific domains
  - Application frameworks – ASC, NGS, ITR
  - Embedded systems – EHS, NGS, ITR

NSF FY 2005 plans in SDP areas include:

- *The CISE emphasis area "Science of Design" (SoD) will promote the scientific study of the design of software-intensive systems. Complex interdependencies strain our ability to create, maintain, comprehend and control these systems. The Science of Design activity seeks to rectify this situation by building a foundation for the systematic creation of software-intensive systems. This foundation will consist of a body of theoretical and empirical knowledge on design, computational methods and tools for design, and a new design curriculum for the next generation of designers.*
- *The special NSF/NASA "Highly Dependable Computing and Communication Systems Research" (HDCCSR) program will continue support for R&D to advance the design, test, implementation, and certification of highly dependable software-based systems. Awardees are able to use a NASA testbed facility to experimentally evaluate their research products on significant real hardware/software artifacts.*

SDP DOE/NNSA SDP

The Advanced Simulation and Computing (ASC) Program is responsible for providing designers and analysts with a high-fidelity, validated, 3D predictive simulation capability to certify the safety, reliability, and performance of the Nation's nuclear stockpile in the absence of physical testing. ASC's SDP R&D is focused on the:

Problem Solving Environment (PSE) program – one of four Simulation and Computer Science (SCS) thrusts involved in developing the ASC computational infrastructure. PSE goals:

- Create a common, usable application development environment for ASC platforms
- Produce an end-to-end high-performance I/O and storage infrastructure to enable improved code execution
- Ensure secure, effective access to ASC platforms across NNSA labs for both local and distributed computing

FY 2004 PSE priorities include:

- Focus on Red Storm software development environment, preparing for initial delivery and the final integrated platform
- Plan and prepare software environments for Purple C and Blue/Gene L. Planning for new platforms is a significant multiyear activity that involves developing and testing software configurations on smaller platforms and scaling up toward the final ultrascale integrated system.
- Continue performance modeling and measurement for increased effectiveness and efficiency of applications and platforms
- Develop high-performance open-source, Linux-based computing environments, targeting capacity computing. Requests for information (RFI) and requests for proposals (RFP) for open source software (OSS) projects are under way.
- Track and test development of Lustre file system for clusters and continue to evaluate alternative file systems
- Continue to evaluate alternatives to the distributed computing environment (DCE) technology, which DOE/NNSA labs have used for authentication, security, and administrative services, due to technology phase-out at the end of calendar year 2005

In FY 2005, DOE/NNSA plans to:

- *Continue the SDP-related work in ASC's PSE program*

SDP DARPA SDP

In FY 2004, DARPA has two programs in the SDP PCA, both within the agency's Information Exploitation Office (IXO). They are:

Software Enabled Control (SEC) – focuses on the design of advanced control systems for innovative vehicles (Unmanned Air Vehicle [UAV], Organic Air Vehicle [OAV], rotorcraft, fighters), including controls at both the vehicle and mission level. SDP-related technology areas are:

- Active state models enabling dynamic prediction and assessment
- Adaptive, dynamic customization of online controls
- Hybrid, multimodal controls
- Open-control platform with reusable (CORBA-based) middleware and tool support

Final demonstrations of small rotorcraft and fixed-wing vehicles are scheduled.

Model-Based Integration of Embedded Systems (MoBIES) – automates the design, construction, and testing of complex embedded systems by exploiting mathematical relationships between the physical platform, the computational platform, and real-world constraints – i.e., "software that is too complex to write manually." Mathematical models are used to meet multiple requirements for application-independent design tools, such as real-time control, network connectivity, fault tolerance, environmental resilience, physical constraints, and component libraries. Resulting tools for designing embedded applications include:

- Intelligent programming tools
- Smart process schedulers
- Communications configuration
- Online resource allocation
- User interfaces
- Automatic code generation
- Automatic verification and validation
- COTS integration

The MoBIES tool integration framework is being demonstrated in avionics and automotive design projects.

DARPA reports no SDP programs in FY 2005.

SDP NIH SDP

In FY 2004, the Center for Bioinformatics and Computational Biology of NIH's National Institute of General Medical Sciences (NIGMS) launches several initiatives:

Establishment of four NIH National Centers for Biomedical Computing – each center will serve as a node for developing, disseminating, and providing relevant training for computational tools and user environments in an area of biomedical computing.

Investigator-initiated grants – intended to foster collaboration with the National Centers. The objective is to avoid competition between "big" and "small" science by encouraging collaborations within the Centers.

Program for creation and dissemination of curriculum materials – designed to embed the use of quantitative tools in undergraduate biology education

The National Centers for Biomedical Computing grants solicitation includes a specific software requirement: Source code supported in this initiative must be shared and users may modify it, creating a pathway to commercialization.

In addition, the FY 2004 grant program of NIH's agencywide Biomedical Information Science and Technology Initiative (BISTI) includes a category for computer science grants. BISTI also has a program in software development and maintenance.

SDP NIST SDP

SDP-related activities at NIST include:

- Health: Informatics, privacy, ubiquitous aids, device controllers, diagnosis aids, decision support, and computer-aided surgery
- Nanotechnology:Computer control, modeling, and visualization
- Information and Knowledge Management: Data and knowledge bases, formal representations, intelligent access, and manipulation and testing tools
- Cybersecurity and Critical Infrastructure: Computer security, encryption, biometrics, monitoring software, and computer forensics as well as precision engineering and calibration for gun-tracing equipment, bullet-proof vests, and other security-related devices
- Weapons Detection: Monitoring devices such as sensors, threat-detection information systems
- Measurement Science: Conformance testing, scientific visualization and models, intelligent measuring tools, and machine learning tools

Within NIST's Systems Integration for Manufacturing Applications (SIMA) program, SDP-related projects include:

Automating Equipment Information Exchange (AEX) – aims to automate equipment design, procurement, and operation through software interoperability. Interoperability issues focus mainly on data.

Digital Library of Mathematical Functions (with NSF) – FY 2004 activities include usability studies and completion of the Web site.

Electronic Commerce for the Electronics Industry (ECEI) – supply chain software interoperability

IT Infrastructure Conformance Testing – working with industry partners on standards, such as ebXML, a version of XML for electronic business, and is partnering with Korea and Europe on testbeds for trying out b2b solutions

Interoperability of Databases for the Structure, Stability, and Properties of Inorganic Materials

Manufacturing Enterprise Integration – involves a testbed and work with industries on software interoperability, including research on automated methods for integrating systems. Research takes an ontological approach, using formal logic, to enable agent technologies and expert systems to automate the process of integrating systems.

NeXus Data Exchange Standard – developing sharable data structures for neutron researchers

Numerical Data Markup Language – developing UnitsML, an XML schema for encoding measurement units

Open Architecture Control – developing key interface standards and associated conformance tests to achieve interoperability of manufacturing control systems architectures with security taken into consideration. This has CIP applications, such as electrical generating plants or hydroelectric dams.

Product Data Standards for HVAC/R

Product Engineering Program – developing a semantically-based, validated product representation scheme as a standard for seamless interoperability among CAD systems and with systems that use CAD data

Standards for Exchange of Instrument Data and NIST Chemical Reference Data

Standards for Physical and Chemical Property Data Interchange

Anthropometric Data Standards (with U.S. Air Force) – seeking accurate 3-D representation of human measurements; an application is cockpit design.

NIST FY 2005 plans in SDP R&D include:

- *Continue SDP-related work in health informatics, nanotechnology, information and knowledge management, cybersecurity and critical infrastructure, weapons detection, and measurement science*
- *Continue SIMA program activities listed for FY 2004*

SDP NOAA SDP

NOAA SDP activities focus on using two software modeling frameworks:

Flexible Modeling System (FMS) – NOAA's Geophysical Fluids Dynamic Lab (GFDL) uses the FMS programming structure to develop atmospheric, ocean, and coupled climate models for climate projection studies.

Earth System Modeling Framework (ESMF) – ESMF, the next-generation community-wide version of FMS, is being worked on by GFDL in collaboration with NASA, NSF's National Center for Atmospheric Research (NCAR), the university community, and NOAA's National Weather Service (NWS).

The challenges of building increasingly interdisciplinary Earth system models and the need to maximize the performance of the models on a variety of computer architectures, especially those using upwards of thousands of processors, necessitates a new program structure. This structure allows physical, chemical, and biological scientists to focus on implementing their specific model components. Software engineers then design and implement the associated infrastructure and superstructure, allowing for a seamless linkage of the various scientific components.

FY 2005 plans include:

- *Continued cooperative ESMF development with NSF and NASA*
- *Establish Development Test Center (DTC) with NCAR and the university community*
- *Provide modeling framework for weather research and atmospheric science*
- *Improve the transition of research to operations*

PARTICIPATING AGENCY

SDP FAA SDP

Safety is the FAA's primary mission. Safety will remain the FAA's top priority as the aviation industry readjusts itself to a world transformed by terrorism and economic challenges. Increasingly, solutions to these challenges depend upon secure and dependable software-based systems and the ability to produce them in a timely, predictable, and cost-effective manner.

FAA work straddles the SDP and HCSS PCAs.(Please see the discussion of FAA activities in the HCSS section.)