Carnegie Mellon University
Software Engineering Institute

Quarterly Update

January – March 1989

Carnegie Mellon University
Software Engineering Institute

Quarterly Update

Published as SEI-89-CDRL-103-1 by the Software Engineering Institute. The SEI is a federally funded research and development center sponsored by the Department of Defense under contract to Carnegie Mellon University.

ESD-TR-89-38
F19628 85C 0003
per form 50

January – March 1989

Director's Overview

During the first three months of 1989, SEI efforts yielded successful results. By releasing products, sponsoring workshops and meetings, and receiving media recognition, the SEI is making substantial headway toward accomplishing our mission of technology transition.

SEI products take various forms, including software, courses, and publications. Since January 1989, the SEI released two prototype systems: the DARK Kernel (p. 11) and the Serpent user interface management system (p. 12). We have delivered such courses as the self-assessment training course (pp. 4, 22), the software inspections course (pp. 5, 22), and the first course offering in the Continuing Education Series (p. 15). Publications of the past quarter included the *Ada Adoption Handbook on Compiler Evaluation and Selection* (p. 10), the *Software Engineering Education Directory* (p. 14), and the *Summary of Technical Operations 1988* (p. 22). These products help us disseminate information about advances in software methods and technologies.

Some SEI events during the past quarter were the Software Capability Evaluation Workshop (p. 5), the Faculty Development Workshop (p. 14), and visits by corporate software steering committees from Martin Marietta and Rockwell International (p. 21). These events brought people together to share information and discuss software issues that are relevant to software practitioners.

Two of the three papers chosen by *IEEE Software* as its best of 1988 were written by SEI authors. (The third was written by a Carnegie Mellon professor.) The award for the best article in the journal went to "Intelligent Assistance for Software Development and Maintenance" by Gail E. Kaiser, Peter H. Feiler, and Steven S. Popovich. This article, which appeared in the May 1988 issue, was based on work done at the SEI by the authors. The first runner-up was "Characterizing the Software Process: A Maturity Framework" by Watts S. Humphrey, director of the SEI Process Program. This paper was in the March 1988 issue of *IEEE Software*.

The SEI also continues to receive favorable media coverage in publications such as *UNIX Today!* (January 23), the *National Journal* (February 5), *Government Computing News* (February 20), the *Chronicle of Higher Education* (February 22), *Federal Computer Week* (March 6), and the March issue of *American Programmer*.

On May 2-4, 1989, we will host the SEI Affiliates Symposium. The theme of this year's symposium is "Software: America's Competitive Edge." Lloyd K. Mosemann, Deputy Assistant Secretary of the Air Force for Logistics, will give the keynote presentation. Participants will have an opportunity to engage in briefings, workshops, tutorials, working sessions, and demonstrations.

In the coming months, the SEI will sponsor several other events, including the Sixth IEEE Workshop on Real-Time Operating Systems and Software (May 11-12, 1989), the 11th International Conference on Software Engineering (May 15-18, 1989), and Software Engineering Education and Training Week (July 17-22, 1989). Future issues of the *Quarterly Update* will report on these events.

Larry Druffel, Director

Table of Contents

Software Engineering Process Program

The Software Engineering Process Program focuses on improving the process of software development. Projects within the program are assessing the actual practice of software engineering in the defense community, training organizations to gain management control over their software development processes, and establishing a network of process groups to improve their software capability.

Software Process Assessment Project

The Software Process Assessment Project assists software organizations in launching effective process improvement programs, characterizes and reports on the software engineering capabilities of defense contractors, and defines priority needs for software process improvement in the defense community.

A final assessment report and a briefing on action recommendations were delivered to Ford Aerospace in February 1989. The report detailed findings from an assessment that project members conducted for Ford Aerospace in December 1988.

In conjunction with staff from the SEI Transition Training group, project members conducted self-assessment training courses at the SEI and at Texas Instruments in Dallas, Texas; and hosted a workshop assessment at the Fifth Annual National Joint Conference sponsored by the National Security Industrial Association (NSIA).

In March 1989 the project gave two self-assessment briefings. One briefing was conducted for Motorola in Phoenix, Arizona. More than fifty people from government and industry attended a second briefing held at the SEI. Affiliates are required to attend a one-day briefing before the SEI will schedule self-assessment training for them.

The first state of the practice report (CMU/SEI-89-TR-1) was published in this quarter. This report provides an overview of the SEI process framework and assessment methodology, describes assessment results obtained so far, and discusses implications of the current state of the practice for customers and suppliers of DoD software.

A report on conducting SEI-assisted assessments (CMU/SEI-89-TR-7) was updated and published in this quarter. This report is addressed to managers and practitioners in organizations interested in having an SEI-assisted assessment of their software process.

The project leader gave an invited presentation on software process assessments at the first annual U.S. West Software Engineering Conference in January 1989 in Denver, Colorado. Approximately 300 senior-level software practitioners and managers attended this conference.

Software Process Development Project

The Software Process Development Project supports improvement in the software engineering process through process groups (SEPGs), guides, workshops, working groups, and training courses. Using data gathered from the Software Process Assessment and the Software Capability Evaluation Projects, the project identifies priority needs for process improvement, selects methods and practices to meet these needs, and works with organizations to adopt appropriate methods and practices.

The third and fourth pilot sessions of the software inspections course were taught to Granite Sentry project personnel at the Air Force Space Command in Colorado Springs in January 1989, and to software practitioners at Computer Sciences Corporation in Washington, D.C., in March 1989. This course, developed in collaboration with the SEI Transition Training group, teaches a method for effectively conducting peer technical reviews of software products, including design and code. It instructs practitioners in cost-effective methods of detecting defects prior to the testing phases of projects.

The project completed a preliminary version of the *Software Engineering Process Group (SEPG) Guide* in January 1989. This report provides guidance on establishing and running an SEPG and will serve as the basis of a future SEI publication.

Software Capability Evaluation Project

The Software Capability Evaluation Project helps DoD acquisition organizations to evaluate the capability of contractors to perform competently on software engineering contracts. The project is improving and transitioning an evaluation method for use in software-intensive acquisitions.

The Software Capability Evaluation Workshop was held at the SEI on March 1-3, 1989. Attending the workshop were 47 participants from 27 organizations. Representatives from 23 of the 24 DoD contractors attending reported that their companies had software improvements underway based on the SEI software maturity framework. The project sponsored the workshop to gather comments on proposed changes to the software capability questionnaire, to collect data on company practices covered by the questions, and to promote dialogue among questionnaire users.

A draft revision of the software capability questionnaire was released for selected review in February 1989. The revision contains proposed changes to format, phrasing, and question content; the changes reflect SEI, industry, and government experiences in field-testing a preliminary version of the questionnaire. The goal of the revision is to improve the clarity of the questions and to increase coverage of software engineering practices.

In January 1989, project members made presentations to the Program Engineering Offices of the Army Communications and Electronics Command (CECOM); and in February 1989, two evaluation teams from CECOM were trained. The project is supporting CECOM, which is using capability evaluations in its acquisition activities.

The project leader gave a presentation on the capability evaluation method in January 1989 to the Los Angeles section of the American Society for Quality Control. Attending were 350 people from 200 companies.

Process Program Reports

January–March 1989

The State of Software Engineering Practice: A Preliminary Report
(CMU/SEI-89-TR-1)

Conducting SEI-Assisted Software Process Assessments
(CMU/SEI-89-TR-7)

For information on how to order reports, see page 26.

Software Engineering Methods Program

The primary objective of the Methods Program is to improve the individual and team productivity of software engineers through the use of modern software engineering technology, that is, modern and emerging tools and methods. To accomplish this objective, the Methods Program identifies valuable technology and promotes its use, including reducing the risks associated with its adoption.

Applications of Reusable Software Components Project

The Applications of Reusable Software Components Project, which began in FY87, is completing its investigation of reusing software components in real-time system development. The project performed a series of software engineering experiments to determine the feasibility of applying reusable components, such as the Common Ada Missile Package, to develop subsystems for real-time applications, such as the Tomahawk Land Attack Missile (TLAM) guidance system.

As part of its redevelopment efforts, the project performed a features analysis of the requirements for the TLAM guidance system and determined that the separation of domain information (functions, data, relationships) from the software architecture was essential to support reuse. In addition, the project defined a set of reusable requirements templates that describe the scheduling, deadlines, timing budgets, and priorities of real-time tasks. The results of these tasks are being used to plan a proposed follow-on project.

Software Development Environments Project

A continuation of the Evaluation of Environments Project, this project is investigating emerging environment technology and its transition to practice. Specifically, the project is studying the problem of tool configuration management and is tracking the evolution of environment architectures.

The tool configuration task involves examining the assembly, integration, and upgrading of an environment that consists of a collection of third-party tools. The task consists of two parts. The first part is to analyze existing techniques for parameterizing tools and mechanisms for source code configuration in terms of their applicability to tool configurations, as well as to model configuration descriptions and analyze their consistency. The second part is to develop, in stages, a proof-of-concept prototype of a support tool that manages consistent tool configurations. Work began this quarter in both areas of this task.

To track the evolution of environment architectures, the project has examined several commercial environments (including the Atherton software backplane, the Rational Ada Environment, and the Sun Microsystem Network Software Environment) as sample environments representative of different technologies. Based on their observations, project members decided to focus on recent advances made by commercial environments in configuration management functionality.

A draft of a chapter on the state of practice of software development environments was completed for a book being published by the American Institute of Aeronautics and Astronautics (AIAA) as a AIAA Progress Series Book on Aerospace Software Engineering.

Also this quarter, project members presented the topic of software development environments at the National Security Industrial Association (NSIA) Conference and at the Open Systems Foundation SIG CASE meeting.

Specification and Design Methods and Tools Project

A proposed follow-on to the Tools and Methodologies for Real-Time Systems Project, this project is examining the use of traditional and formal methodologies in the specification of embedded applications.

Project members are conducting an evaluation of the following five software specification methodologies: Ward/Mellor, Hatley/Pirbhai, Harel, Extended Systems Modeling Language, and Systems Engineering Methodology. This evaluation involves using each method to specify a small-scale problem and evaluating the performance of each method based on criteria developed by the Tools and Methodologies for Real-Time Systems Project. The principal developers of each method have agreed to review the resulting specifications. The project has completed a preliminary evaluation of the Ward/Mellor, Hatley/Pirbhai, and Harel methods.

Also completed this quarter was a classification, by a resident affiliate from SYSCON, of the SYSCON design method, as well as those methods commonly known as Booch, Buhr, and Pamela. A report on this classification is being written.

The project completed an in-depth study of a formal method, Temporal Logic, this quarter. Project members specified an example problem using the method and verified the consistency of the temporal logic statements with a tool set developed at Carnegie Mellon University. This study provided insight on the expressive power of a single formal method, a detailed understanding of the application of a single formal method, and experience using tool sets designed to support a formal method.

Project members are characterizing the formal methods Communicating Sequential Processes (CSP), Temporal Logic, and a Calculus of Communicating Systems (CCS), which are applicable to the specification of concurrent systems. The project will produce a taxonomy of characteristics, describe how each method fits the taxonomy, and present the strengths and weaknesses of each method.

Software Architecture Design Principles Project

Initiated in October 1988, the Software Architecture Design Principles Project is describing basic design elements used in the description, analysis, and development of software systems. The goal of this project is to develop a fundamental understanding of structures for the software architecture level of design.

Project members are analyzing several software system architectures to identify fundamental design elements and composition operations. This is an iterative process that involves analyzing an existing example system and reformulating its design in terms of a fundamental set of design elements, refining the common set of design elements to reflect lessons learned from the example, and repeating the process on other example systems. To date, the project has examined more than 35 example systems. The immediate recipients of these results will be research and development groups working on architectures in specific application domains and the research community in software engineering and programming languages. In addition, project members are also studying how other engineering disciplines deal with comparable problems. During this quarter, a presentation of the results was videotaped, and a report is being written.

Software Process Modeling Project

Transferred from the Software Process Program this quarter, the Software Process Modeling Project is investigating techniques for modeling the software maintenance process.

Project members have been working on models for post-deployment software support (PDSS) processes of the Navy F-14 aircraft and the Air Force F-16 aircraft. They have compared their model of the F-14 PDSS process as described by Navy regulations with the actual process in use, and reported their results to the Navy this quarter.

During this quarter, project members refined the detailed structure and logic and the interconnections of the F-14 process model and developed narrative descriptions of model components in preparation for a thorough verification walk-through with the Pacific Missile Test Center planned for April 1989.

Project members published *Software Process Modeling: Principles of Entity Process Models* (CMU/SEI-89-TR-2). This paper will also be presented at the 11th International Conference on Software Engineering in May 1989 and published in the proceedings. The project also completed a position paper for their participation in a panel session at the same conference.

Project team members organized and chaired a full-day session on "Software Engineering Processes: Models and Analysis" at the 23rd Annual Hawaii International Conference on System Sciences in January 1989. A paper based on lessons learned from the F-14 modeling effort, *Software Process Modeling: A Case Study*, was presented at the conference and published in the proceedings.

Methods Program Reports

January–March 1989

Software Process Modeling: Principles of Entity Process Models
(CMU/SEI-89-TR-2)

For information on how to order reports, see page 26.

Software Systems Program

The goal of the Systems Program is to improve the development of real-time distributed systems by integrating software engineering with systems engineering and reducing the risk of new technology.

Real-Time Embedded Systems Testbed Project

The Real-Time Embedded Systems Testbed (REST) Project is collecting, classifying, generating, and disseminating information about software development for real-time embedded systems.

The project completed development of the Inertial Navigation System (INS) under the DEC VAXELN system in January 1989 and successfully completed a port of the INS to an MC68020 system in March 1989. The INS was selected by the project as a typical real-time application and is being used to provide empirical evidence that Ada can be used for the design and implementation of time-critical applications. It is also intended to demonstrate the practical application of real-time scheduling theory, generate additional issues to investigate, provide a context for real-time experimentation, and provide a software engineering artifact for real-time programming in Ada.

The first version of the *Ada Adoption Handbook on Compiler Evaluation and Selection* (CMU/SEI-89-TR-13) was published in March 1989. This document provides specific guidance on how to select an Ada compiler. It is a distillation of the experiences and expertise of the SEI and other organizations in selecting compilers.

The first use of the real-time embedded systems testbed by affiliate companies was conducted on January 17-18, 1989, when representatives of General Electric and Calspan visited the SEI. Approximately 75 benchmark performance tests were run; and, with one minor exception, all tests were completed successfully. In addition to determining the results of their tests, the visitors became sufficiently familiar with the testbed that they now can use it from their locations, which they plan to do.

Project members continued to develop and experiment with the Hartstone benchmarks. Several single-task and multi-task versions of Hartstone benchmarks were run. Additionally, a version of a Hartstone benchmark that uses rate-monotonic priority assignment was written and successfully run.

The project finalized the configuration for the Real-Time Computing Network (RTCN). The RTCN is a fiber-optic system which is being provided to the SEI by IBM. The RTCN will provide the testbed with a distributed target system for experimentation and analysis.

Real-Time Scheduling in Ada Project

The Real-Time Scheduling in Ada Project is demonstrating how to design and implement real-time systems using analytic scheduling algorithms.

The implementation of the scheduling algorithms for the VADS/UNIX runtime system was completed, and the modifications performed under VADS/ULTRIX were ported to the VADS/MC68020 runtime system. The implementations will be used to empirically evaluate the theory by running various test cases.

The implementation of the avionics case continued. The test case is being developed in association with the Naval Weapons Center at China Lake and IBM Owego. This test case will be used to demonstrate that the timing and performance requirements of a uniprocessor avionics system can be met in Ada.

The specification of the scheduling algorithms was completed this quarter and will be documented in a forthcoming technical report.

A schedulability analysis of the INS system developed by the REST Project was initiated, and the results were presented in a paper that will appear in the *SEI Annual Technical Review*, which contains technical articles by SEI staff.

The project agreed to provide support to the IEEE FutureBus+ effort by having a representative on the Real-Time Working Group. The real-time scheduling algorithm technology is being incorporated into the next generation IEEE FutureBus+ standard, a backplane bus being developed as part of the Next Generation Computer Resource initiative of the Navy.

Distributed Ada Real-Time Kernel Project

The Distributed Ada Real-Time Kernel (DARK) Project is developing a prototype kernel that supports distributed targets and that will be freely available, tailorable, and extensible.

The first implementation of the Kernel, including the *Kernel User's Manual* (CMU/SEI-89-UG-1) and all other necessary documentation, was completed in February 1989. The Kernel was approved for unlimited public distribution by the SEI Joint Program Office and is now available to acceptor sites as an alpha release version. The Wichita State University and Boeing Military Airplane jointly signed an agreement with the SEI, becoming the first acceptor site. Lockheed Missiles and Space became the second acceptor site, and other organizations are expected to follow. (For more information on DARK transition activities, see page 20.)

The project is modifying the INS developed by the REST Project to use it with the Kernel. Because the Kernel has its own scheduling mechanisms, the dispatcher task of the INS must be removed. The INS will be used to conduct performance testing of the Kernel.

User Interface Prototyping Project

The User Interface Project is developing Serpent, a user interface management system that separates user interface concerns from application concerns, which allows the application of a system to be media independent.

The alpha version of Serpent was completed this quarter and released to acceptor sites. So far, Serpent has two acceptor sites: Jet Propulsion Laboratory (JPL), and Westinghouse Research and Development Center. JPL is interested in testing Serpent for use in the Deep Space Network System, and Westinghouse will use Serpent in their prototyping efforts.

Serpent was approved by the SEI Joint Program Office for unlimited public distribution. The software and documentation of Serpent Alpha Release 0.5 are available to all interested organizations.

Software for Heterogeneous Machines Project

The Software for Heterogeneous Machines Project is developing tools and a methodology to support applications running on networks of different processors executing concurrent, communicating tasks. The heterogeneous machines targeted by this project have general-purpose processors, special-purpose processors, memory boxes, and switches that can be configured in arbitrary logical networks.

Project members completed the initial version of the automatic data transformation and dynamic reconfiguration features of the Durra language that the project developed. The automatic data transformation feature transforms data being sent between task ports and eliminates differences between sending and receiving port data types. The dynamic reconfiguration feature supports changes in the structure of an application (processes and data queues) in response to dynamic runtime conditions. Configuration states are "stacked" to allow applications to revert to a previous configuration.

An initial version of the application debug/monitor task was also developed this quarter. This task provides the application developer and user with the ability to exercise interactive control over the application and to monitor the progress of the application.

Two technical reports and a journal article by project members were published during this quarter:

- *Performance and Reliability Enhancement of the Durra Runtime Environment* (CMU/SEI-89-TR-8) is a study of fault tolerance issues and techniques in distributed systems. It presents a design for a distributed implementation of the Durra runtime system to enhance reliability of the applications.
- *Command, Control, Communications, and Intelligence Node: A Durra Application Example* (CMU/SEI-89-TR-9) describes the results of the first phase of a joint effort with TRW Defense Systems Group to demonstrate the language and methodology in a C3I application.
- "Developing Applications for Heterogeneous Machine Networks: The Durra Environment" appears in the March 1989 issue of *Computing Systems*.

Human-Machine Interaction Considerations for Interactive Software
(CMU/SEI-89-TR-4)

Performance and Reliability Enhancement of the Durra Runtime Environment
(CMU/SEI-89-TR-8)

C3I: A Durra Application Example
(CMU/SEI-89-TR-9)

Ada Adoption Handbook: Compiler Evaluation and Selection
(CMU/SEI-89-TR-13)

Real-Time Scheduling Theory and Ada
(CMU/SEI-89-TR-14)

Kernel User's Manual, Version 1
(CMU/SEI-89-UG-1)

Kernel User's Manual, Version 1, Appendix A: Ada Code
(CMU/SEI-89-UG-1)

Serpent, A User Interface Management System Overview, Version 1
(CMU/SEI-89-UG-2)

SEI Serpent SADDLE User's Guide
(CMU/SEI-89-UG-3)

SEI Serpent SLANG Reference Manual
(CMU/SEI-89-UG-5)

SEI Serpent Application Developer's Guide
(CMU/SEI-89-UG-6)

For information on how to order reports, see page 26.

Education Program

The primary objective of the SEI Education Program is to increase the number of highly qualified software engineers by rapidly improving software engineering education throughout the education communities of academia, government, and industry. To accomplish this objective, the program's projects focus on accelerating the development of software engineering programs in academia and on enhancing opportunities for the continuing education of practitioners.

Graduate Curriculum Project

The Graduate Curriculum Project is developing a model curriculum, promoting the creation of graduate degree programs in software engineering, and increasing the amount of software engineering content in master's degree programs in computer science. The project holds a biannual Faculty Development Workshop and an annual Conference on Software Engineering Education as two of the mechanisms for disseminating its work to the education community.

The fifth SEI Faculty Development Workshop (FDW) was held in Scottsdale, Arizona, on January 6-7, 1989. This workshop provided the SEI with an opportunity to distribute educational materials to educators in academia, industry, and government. The workshop was very well received by the 90 participants, 60% of whom were new to the FDW this year. Discussion topics included a framework for viewing software specification, the SEI Master of Software Engineering curriculum, the place of technical writing in software engineering programs, and the plans for a curriculum module in that area. Participants also received a module user's guide.

Six curriculum modules were released at the FDW. Curriculum modules, which explain a software engineering topic, are designed for use by instructors in planning and developing courses. Each module includes an annotated outline and bibliography, along with advice about teaching the material. The new curriculum modules (p. 17) were written by SEI visiting scientists, who are college and university faculty who work at the SEI, usually for one to three months. Some modules are revisions or updates of earlier work.

The State of Kansas recently approved the software engineering graduate program at The Wichita State University. Wichita State, the first SEI graduate curriculum test site, based the content of their program on SEI curriculum modules.

The third annual *SEI Software Engineering Education Directory* (CMU/SEI-89-TR-10) was published during this quarter. The most comprehensive survey of its kind, the directory summarizes undergraduate and graduate courses in software engineering taught at colleges and universities throughout the U.S. and Canada.

Undergraduate Software Engineering Education Project

The Undergraduate Software Engineering Education Project concentrates on increasing the number of software engineering concepts taught in undergraduate computer science degree programs.

The project released a document of educational material, *Software Maintenance Exercises for a Software Engineering Project Course* (CMU/SEI-89-EM-1). This report provides an operational software system (the Documented Ada Style Checker) of 10,000 lines of Ada code and several exercises based on that system. Concepts such as configuration management, regression testing, code reviews, and stepwise abstraction can be taught with these exercises. Macintosh or PC diskettes containing the code and documentation that accompanies the report are also available.

Video Dissemination Project

The purpose of this project is to increase the availability of software engineering education in academia and industry. The project helps to introduce courses and to prepare educators to offer these courses by producing videotaped lectures and support materials for recommended courses, and by providing tutor orientation and consultation services.

Through its Academic Series, the project cooperates with academic institutions to produce and disseminate graduate courses consistent with curriculum recommendations of the Graduate Curriculum Project and the Carnegie Mellon Master of Software Engineering program.

A course in the Academic Series, Software Verification and Validation is being presented in the SEI video studio to 21 Carnegie Mellon students. Videotapes of the lectures are distributed to the University of New Mexico, East Tennessee State University, and California State University at Dominguez Hills, where the course is offered for credit. Faculty members at these universities, who act as tutors for the course, attended a one-day orientation session at the SEI in January 1989. A professor from the University of California at Santa Barbara visited the SEI to present the course lecture on symbolic execution.

A second course, Software Project Management, is being offered in videotaped form at California State University at Dominguez Hills and at Jacksonville State University. The initial offering of this course was made to Carnegie Mellon students during the summer of 1988. It was presented by a visiting scientist from The Wichita State University, an SEI academic affiliate. In January an orientation session for the course tutors was conducted at Wichita State.

East Tennessee State University has participated in the Academic Series for three semesters. One of their faculty is now prepared to teach a previous offering, Formal Methods in Software Engineering, on his own. One goal of the Academic Series is to help educators begin teaching new courses quickly and to enable them to continue to offer these courses on their own in the future.

Through its Continuing Education Series, the project provides government and industry with courses that offer practitioners prerequisite education for the Academic Series, offer graduate-level content in noncredit form, and keep practitioners apprised of the state of the art in software engineering technology.

In this series, the first of three mini-courses in Software Project Management was presented in February 1989. Twenty-one students enrolled, with representation from the following organizations: Hanscom Air Force Base (AFB), Randolph AFB, Scott AFB, Naval Surface Warfare Center, Space and Naval Warfare System Command, McDonnell Helicopter Company, Westinghouse Electric, Hughes Aircraft, Texas Instruments, IBM, General Dynamics, and the SEI.

Advanced Learning Technologies Project

The Advanced Learning Technologies (ALT) Project is using advanced hardware and software technologies to teach code inspections, a formal review process that identifies defects in software code. These technologies provide self-paced learning while preserving the interactive and adaptive aspects of classroom-style, group-paced training. Interactive, technology-intensive training can be distributed economically to a broad audience, providing a remarkably effective learning experience.

Project members began porting the code inspection prototype to the Digital Video Interactive (DVI) system. The work included digitizing audio sequences and images for the DVI system. Animation was successfully digitized and timed with its audio file. The prototype graphics software subsystem was ported to the DVI system, and the graphics art work was begun.

The SEI has become a beta site for Intel DVI technology. Project members demonstrated the port of the ALT prototype to the DVI system at the Intel exhibit booth at the Microsoft CD-ROM Conference in Anaheim, California, in March 1989.

A paper on development issues was presented at a seminar entitled, "Use and Development of Interactive Multi-Media in Higher Education." This was a national seminar funded by Digital Equipment Corporation and organized by Carnegie Mellon's Center for Design of Educational Computing (CDEC). Held in Kissimmee, Florida, on February 20, 1989, the seminar was attended by developers, vendors, and educators.

Software Engineering Video Network Function

The Software Engineering Video Network (SEVN) Function provides the facility, equipment, and technical expertise to support the audio/video production requirements of the SEI. The SEVN studio is primarily used to tape courses for the Video Dissemination Project, as well as for the videotaping and editing activities of the Advanced Learning Technologies Project. The SEVN facilities are also used by other SEI programs and by Carnegie Mellon.

An Educational Communications (EDUCOM) Conference was held at the SEVN facility in February 1989. Presentations included a one-hour program highlighting new computerized authoring environments, which had been taped at the studio earlier.

An audio interview about the Computer Emergency Response Team Coordination Center was taped at the request of the Carnegie Mellon Public Relations Department. This interview was provided to ABC Radio News in February 1989.

Introduction to Software Design (Revision)
(SEI-CM-2-2.1)

Unit Testing and Analysis (Revision)
(SEI-CM-9-1.1)

Software Metrics (Revision)
(SEI-CM-12-1.1)

Introduction to Software Verification and Validation (Revision)
(SEI-CM-13-1.1)

Software Requirements
(SEI-CM-19-1.0)
[Supercedes *Requirements Specification Overview* (SEI-CM-1-1.3)]

Formal Verification of Programs
(SEI-CM-20-1.0)

These curriculum modules were written by visiting scientists.

Education Program Curriculum Modules

January–March 1989

SEI Software Engineering Education Directory
(CMU/SEI-89-TR-10)

Software Maintenance Exercises for a Software Engineering Project Course
(CMU/SEI-89-EM-1)

Education Program Reports

January–March 1989

For information on how to order curriculum modules and reports, see page 26.

Ada & STARS Support

As Ada matures and its use becomes more common within the mission-critical computer resource (MCCR) community, users will be faced with the problems of adopting more disciplined approaches to software engineering. The goals of the Ada and STARS Support effort are to remove technical and managerial impediments to the adoption of Ada, to support the DoD STARS (software technology for adaptable, reliable systems) Program in technology development and transition efforts, and to explore the advantages and disadvantages of new software engineering approaches and paradigms made possible by Ada language features.

STARS Shadow Project

The SEI is providing specific support to the STARS Program Office for the execution of shadow projects. The purpose of the shadow projects is to run in parallel with existing development efforts (known as parent projects) but not to interfere with these efforts. These shadow projects apply technology and approaches that potentially offer dramatic improvements in aspects of software development.

SEI project members have contributed significantly to an Eglin Air Force Base project to redesign the Maverick air-to-ground missile seeker from infra-red sensing to millimeter-wave radar. Hercules Defense Systems, the contractor for the shadow project, is writing, in Ada, the control software for the embedded 80286/87 processor set. SEI project members are currently working with Hercules to develop an object-oriented architecture for the design of the software. In this application domain, project members are developing a preliminary set of object-oriented architectural models, which can be assembled to provide specific mission controllers.

Domain-Specific Software Architectures Project

The Domain-Specific Software Architectures (DSSA) Project provides solutions to the design problems that characterize an application domain. The project is currently focusing on the domain of command, control, communications, and intelligence (C3I) systems. The purpose of this project is to increase the use of modeling in the development of system architectures. Project members are working closely with the Granite Sentry Project at Cheyenne Mountain to provide software engineering and Ada design support to the Granite Sentry development. Specifically, the project is identifying recurring problems and developing model solutions for C3I systems. Recurring problems are those that appear in multiple instances within a given system, or those that appear substantially unchanged from system to system.

Project members have identified the recurring problem of message translation and validation, and have worked with Granite Sentry to determine the requirements for various messages. Project members modeled a solution to this problem and expressed it in a form that is easily applied to any of the message types. To support Granite Sentry's implementation of this solution, project members developed software templates so that less-skilled programmers can easily apply the solution to any message type.

In March 1989, the first critical design review was held to examine the full implementation of a typical message received from a sensor. The full implementation provides early estimates of system performance for the full message set and serves as a pattern for implementing the remaining messages.

Binding of Ada and SQL Project

The Binding of Ada and SQL Project, initiated at the request of the Ada Joint Program Office (AJPO), has investigated the problem of binding the Ada programming language with the Structured Query Language (SQL) database language. The solution to this problem is the specification of an interface, the SQL Ada Module Extensions (SAME), that permits an application program written in Ada to access and manipulate data controlled by a database management system (DBMS) using the SQL database language.

To support the implementation of the SAME methodology, project members developed more than 1500 lines of Ada code known as the SAME standard packages. Project members are porting these packages to insure that this code will run correctly on those platforms most common to DoD systems. The platforms include the following (listed by hardware/operating system/Ada compiler/database):

- Sun/UNIX/Alsys/Ingres
- VAX/UNIX/Verdix/Ingres
- Zenith 248/MS-DOS/Alsys/Oracle
- IBM/MVS-CICS/Intermetrics/DataComm
- IBM/MVS-TSO/Intermetrics/DB2

A SAME Design Committee meeting was held in March 1989. The SAME Design Committee is composed of organizations interested in Ada and SQL binding, including compiler vendors, DBMS vendors, Army projects and agencies, and expected users. Periodically, the committee reviews the project to ensure that it satisfies the requirements of the user community. During their March meeting, committee members discussed building an automated approach. The automated approach will provide a tool-based approach for binding Ada and SQL, which will relieve the application developers from many of the activities that are currently done manually.

Project members refined the instructions for the manual implementation of the SAME methodology, and finalized a report on these guidelines.

Study and Analysis for Ada 9X Project

The Ada Joint Program Office has decided that a revision to the *Ada Reference Manual* (ANSI/MIL-STD-1815A) is required to maintain it as a current ANSI standard. This revision process is commonly referred to as Ada 9X. Because several SEI technical staff members possess extensive expertise in the design and development of Ada, the SEI will provide an organizational framework to help put proposed changes in perspective and guide revision activities.

This project, which began in February 1989, is currently examining areas in the Ada standard to consolidate the technical issues that must be resolved, analyzing the rationale underlying original design decisions that may need to be changed, and discussing the benefits and disadvantages of possible revision approaches.

Ada & STARS Support Reports

January–March 1989

An Object-Oriented Solution Example: A Flight Simulator Electrical System (CMU/SEI-89-TR-5)

For information on how to order reports, see page 26.

Technology Transition Program

The Technology Transition Program is the focal point for SEI transition efforts. The program works with other SEI programs to match problems and solutions in the DoD software community.

Technology Application Function

The Technology Application Function provides a link between DoD mission-critical areas and ongoing SEI technology evaluation and development efforts. The initial focus is on specific MCCR application domains. The goal is to identify technologies useful in a particular application domain and to assist in their maturation by working with specific clients to provide product transition and direct support.

The Distributed Ada Real-Time Kernel (DARK), developed by the DARK Project, was sent to the first acceptor site on February 28, 1989. The acceptor site is a joint site consisting of the Wichita State University and Boeing Military Airplane Company. The transfer follows the terms of an agreement signed by all three parties. Lockheed Missiles and Space Company became the second acceptor site. The transition process will culminate in the selection of one or more product affiliates, who will commercialize the Kernel. *DARK Update*, a newsletter for informing members of the real-time systems community about the status of the Kernel, summarizes the project's technical and transition activities.

During this quarter, a member of the Domain-Specific Software Architectures Project joined Technology Applications to assist in disseminating DSSA technology. A domain-specific software architecture (DSSA) is a set of models that are the basis for the specification, design, and implementation of systems in an application domain. Work on DSSAs at the SEI began in the domain of flight simulators and training devices. Efforts continued in support of pattern-based models in general and a model for training systems developed by the SEI in particular. The USAF System Program Office (SPO) for Training Systems has entered into an agreement with the SEI to transition lessons about design, development practices, and documentation to programs expressing interest in DSSA technology. These programs include the MV-22 simulator, the C-17 simulator, and training systems for the space station.

The *Proceedings of the Workshop on Executive Software Issues* was published as an SEI technical report (CMU/SEI-89-TR-6). It describes the major software issues facing the U.S. from the point of view of executives of industry, government, and academia. The report provides source material for the Air Force Contract Management Division (AFCMD) Joint Executive Conference (JEC) session on software issues to be held in May 1989 as a step toward developing a national strategy for software.

The Affiliate Relations Function establishes and maintains SEI relationships with industry organizations, providing access to SEI information through *Bridge* mailings, telephone contact, special meetings, and the annual Affiliates Symposium. Affiliate Relations also negotiates and places, with the cooperation of the SEI Joint Program Office, government and industry resident affiliates at the SEI. The SEI Education Program manages academic affiliates and visiting scientists from academia.

The SEI hosted corporate software steering committees from Martin Marietta (Information Systems Group and Information Technology Institute) and Rockwell International. These were two in an ongoing series of meetings with software steering committees from major defense contractors. The SEI has now hosted seven of the top twenty companies. Because these groups have a broad perspective, and extensive experience as high-level managers and technical staff, they represent important allies within the contractor community.

A technical exchange meeting was held with Hughes Aircraft's Radar Systems Group to discuss their interest in collaborating on real-time Ada projects at the SEI. Further interactions have been planned.

Nine new industry affiliates signed information exchange agreements during this quarter, and three new resident affiliates joined SEI projects. The chief scientist of Hughes Aircraft Company, Information Systems Division, joined the SEI as a resident affiliate and began working with the Software Capability Evaluation Project in the Process Program. Two other resident affiliates began working with the User Interface Prototyping Project in the Systems Program, one from the National Security Agency and one from the Naval Air Development Center. The resident affiliate from SYSCON, who was working on the Specification and Design Methods and Tools Project in the Methods Program, concluded his term of residency. As of March 31, 1989, fourteen resident affiliates are working at the SEI, six from industry, and eight from the Services and government agencies.

The program announcement and registration materials for the 1989 SEI Affiliates Symposium were prepared and mailed. The symposium will take place in Pittsburgh on May 2-4, 1989. The theme of this year's symposium is "Software: America's Competitive Edge." The symposium will highlight both the state of software engineering practice and the state of software engineering technology. (For more information on the Affiliates Symposium, see the Director's Overview on page 2.)

Transition Training Function

The Transition Training Function provides training development and delivery to support transition activities of SEI technical projects. In addition, the function provides training on managing the organizational change associated with technology transition, and consulting services to customers on software engineering training issues.

A course on process self-assessment underwent pilot testing at Texas Instruments and at the SEI. The course is under revision by staff from the Software Process Assessment Project and Transition Training. Staff members participated in a briefing for organizations that may send assessment teams to the self-assessment training course in May 1989.

In conjunction with the Software Process Development Project, a course on software inspections was developed and presented to the Granite Sentry Project at Air Force Space Command and to the Defense Systems Division of Computer Sciences Corporation.

A course on managing technological change was presented in January 1989 to SEI personnel and invited guests. The course enables internal technology transition personnel to better prepare for and manage the organizational changes incurred by the adoption of new technology. A presentation on managing technology transition was made to the senior management team for Contel Corporation's Computer Technology Center. As a result, the SEI has been invited to teach the course at Contel in April 1989.

Presentations on the applications of software engineering techniques to training systems design were made at the annual meeting of the Association for Educational Communications and Technology. A presentation on configuration management training was made at the annual meeting of the Technology and Innovation in Training and Education Conference. These presentations represent an effort to transition software engineering technology to other disciplines that support software development.

Information Management Function

The Information Management Function develops, edits, publishes, and maintains information relevant to the mission and accomplishments of the SEI.

Several publications were issued this quarter. *Bridge* is the SEI magazine, and the *Summary of Technical Operations: 1988* is an annual report on SEI accomplishments. *DARK Update is a* newsletter produced jointly with Technology Application and the Distributed Ada Real-Time Kernel (DARK) Project. A brochure on the Advanced Learning Technologies Project includes an article discussing the project, a list of academic affiliates, and a one-page overview of the SEI. A brochure and registration form describes the Software Project Management course, the first in a series of Continuing Education courses developed by the SEI Education Program. Product literature on the SEI Software Tools Database describes the process by which the SEI will provide verified, thorough information on available software tools to enable objective comparison of tools.

The SEI library added 125 book titles and 6 periodicals to its collection and 250 documents to its database during this quarter. The collection now comprises 2300 book titles and 220 periodicals. The database contains information about 2800 documents germane to software engineering. The library collection is available through the OCLC interlibrary loan network, which provides access to 3400 participating libraries in the U.S. The SEI library also distributes SEI technical reports to 27 libraries, 10 of which send their organization's reports to the SEI library.

After successful exploratory use in 1988, Information Management began production use of the Interleaf Technical Publishing System on Apple Mac II and Sun 3/60 workstations. Several complex documents were produced and used to derive structural templates for basic report types. Initial tests successfully incorporated text and graphics into Interleaf from several applications with output in tiff, pict, and rtf formats.

Empirical Methods Function

The Empirical Methods Function supports transition management of technology projects by providing market research methods and materials, conducting surveys, and evaluating or validating technologies produced by SEI projects.

In collaboration with staff from the Process Program and American Institutes for Research, work began this quarter to improve data gathering instruments for the Process Program's process assessment activities. The initial result of this effort is a validation plan for the Process Program's effort to refine the contractor assessment questionnaire. In addition, a protocol analysis and pilot test are being developed to ensure the reliability of the assessment instrument.

A pilot effort is underway to prepare a "lessons learned" study of the Real-Time Scheduling in Ada Project. Interviews are conducted bi-weekly with project staff and external contacts to document project events and impacts.

Technology Transition Program Reports

January–March 1989

Proceedings of the Workshop on Executive Software Issues
(CMU/SEI-89-TR-6)

For information on how to order reports, see page 26.

Computing Facilities

Computing Facilities is responsible for the provision, operation, and support of the SEI computing environment. Because of the variety of individual and project needs presented by the SEI population, a heterogeneous environment is clearly required. The challenge is to provide an appropriate mix of hardware, software, and support services to support the SEI mission. Additionally, Computing Facilities provides computing acquisition support (including consultation, negotiation, and coordination) and assessment of new technologies in the marketplace.

All SEI UNIX workstations have been converted from individual file systems to a network file system (NFS). This technology was initially developed by Sun Microsystems, and implemented by Digital Equipment Corporation in ULTRIX, their UNIX derivative. NFS provides a common view of the entire SEI file system, from the more than 100 UNIX workstations at the SEI. NFS simplifies configuration management, backups, and system maintenance of UNIX systems, and encourages sharing data and computer resources. NFS, along with the already existing VMS VAXclusters and AppleShare servers, provide convenient back-end file services to all SEI computer users.

Computing Facilities is integrating the AppleTalk and Ethernet networks so that the two networks can work together to improve the overall computing environment. Kinetics FastPath gateways were installed between Ethernet and AppleTalk, which enabled three other advances this quarter. Macintosh users now have a better file server with more disk space on a more powerful system. Macintosh users can also log into VMS or UNIX systems without using a dedicated serial line, allowing users to have multiple windows logged into the same or different host systems. Finally, Macintosh users can now use the Digital LPS40 laser printer, which prints 40 pages per minute.

CERT Coordination Center

The purpose of the Computer Emergency Response Team Coordination Center (CERT/CC) is to supplement existing mechanisms by which informally organized experts deal with and prevent computer emergencies. The CERT/CC at the SEI supports two different communities: Internet users and producers of technology that is available on the network, such as UNIX and networking software. The services provided by the CERT/CC generally fall into these categories:

- Provide a dependable and trusted 24-hour point of contact for security issues.
- Maintain a reliable and highly secure repository of information for team members.
- Provide rapid communication during emergencies.
- Maintain close ties with research in the area of trusted systems to improve the security of existing systems.
- Raise constituents' awareness of security issues and assist individual organizations in improving the security of their systems.

To organize the CERT/CC, project members installed a phone system that answers inbound calls at all times; installed computer systems (machines, procedures, and software) to interact with the client communities and to support a database of information on CERT members, known vulnerabilities, fixes, configuration information, and security related events; and contacted several hundred people in the client communities to develop working relationships. In cooperation with the National Institute of Standards and Technology, the CERT/CC also held working sessions with several government agencies to form a federation of cooperating organizations to deal with computer security problems.

Since its inception in November 1988, the CERT/CC has assisted client communities in their efforts to deal with security events. In each case, the CERT/CC notified the appropriate agencies of the event, and worked with members of the client communities to identify the source of the problem and facilitate its resolution. In cases where a major security vulnerability exists, the CERT/CC works with vendors and the Network Information Center to inform network users of the problem, provide information on how to protect their systems, and assist in distributing corrections when they become available.

For More Information

How to Order SEI Publications

To order an SEI publication, send a written request accompanied by a preaddressed mailing label to:

Software Engineering Institute
ATTN: Publications Requests
Carnegie Mellon University
Pittsburgh, PA 15213-3890

How to Get the SEI Magazine

To be added to the *Bridge* mailing list, write to:

Software Engineering Institute
ATTN: Bridge Mailing List
Carnegie Mellon University
Pittsburgh, PA 15213-3890

How to Become an SEI Affiliate

For information on opportunities for affiliation, contact:

Software Engineering Institute
ATTN: Mark Coticchia
Carnegie Mellon University
Pittsburgh, PA 15213-3890
(412) 268-6138
ARPANET: mec@sei.cmu.edu

How to Get Additional Information

For further information about the SEI, contact:

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
(412) 268-7700