In This Issue...

- **A Synopsis of Seven Measurement Initiatives**
- **Requirements Engineering Environment (REE) An Executable Prototyping Toolset for Requirements Engineering**
- **DACS Newsletter Editorial Board**
- **Visit the DACS Booth at the Software Technology Conference '95**
- **Criteria for Managing Technical Reviews and Audits**
- **Workflow Management Using OASYS**
- **19th Annual Software Engineering Workshop**
- **Software Engineering Events**
- **DACS Technical Reports Available On-Line**
- **About the Newsletter**
- **The 10th Knowledge-Based Software Engineering Conference (KBSE-95) Call for Papers, Demos, Panels, and Tutorials November 12-15, 1995 Boston, Massachusetts, USA**

Register Now for Software Tech News
There are a number of initiatives and programs underway in the DoD. Some of them have been in place for some time while others are of recent vintage. Our objective is to provide a periodic status of these programs in future issues of this newsletter. Submissions describing other measurement programs are encouraged and welcome. Please address your submissions to the DACS Newsletter Editor.

Rome Laboratory Software Quality Technology Transfer Consortium (SQTTC)

This activity was reported on in the Winter 1994 issue of the DACS Newsletter. The initiative is an attempt to transfer the technology known as the Software Quality Framework. Developed between 1980 and 1985 this framework is based on the description of thirteen attributes of software called software quality factors. Its genesis is the early work done by Barry Boehm, Jim McCall, and others to identify quality attributes. The taxonomy is based on the association of various criteria such as "anomaly management" with each factor. Each criterion is measured through various metrics or elements that are associated with the products and various phases of the development life cycle. The consortium has three active member companies: Northrop Grumman, Hughes Canada, and SoHaR Inc. Northrop Grumman is applying the framework to the Joint Surveillance Target Attack Radar System (Joint STARS) program while Hughes to the Canadian Air Traffic Control System. Lockheed Fort Worth Company is in the process of joining the consortium. The consortium is seeking new member companies. Andrew Chruscicki, founder and manager of the Consortium, has moved to assume the role of Technical Director. John Marciniak, Technical Director of the DACS, has recently been brought on board to assist Mr. Chruscicki for day-to-day management. Please direct your inquiries to Mr. Thomas McGibbon, dacs-director@dacs.dtic.mil.

MIL-Handbook-Practical Software Measurement

This initiative is to develop a measurement guidebook that can be used in conjunction with the soon to be released Military Standard 498, the revision to DoD-STD-2167A (DoD Systems Software Development). The guidebook is titled MIL-Handbook-Practical Software Measurement. The primary objective of the guidebook is to provide practical software measurement implementation guidance to DoD Mission Critical Computer Resources (MCCR), Command, Control, Communications and Intelligence (C3I), and Automated Information System (AIS) program acquisition organizations in support of MIL-STD-498. The effort is sponsored by the Joint Logistics Commander's Computer Resources Management Committee, in short the JLC-CRM. The JLC-CRM also sponsors the 498 revision to 2167A. The guidebook will be published in its initial version this summer. It will concentrate on the "measurement
principles" that program managers and engineering personnel should be aware of and utilize. An accompanying case study is under preparation that will illustrate the principles of the guidebook. The point of contact for this effort is:
Jack McGarry
Naval Undersea Warfare Command
(401) 841-3834, mcgarry@ada.npt.nuwc.navy.mil.

SEI Core Metrics Program

The SEI has developed a set of guidebooks that define a basis set of core metrics as well as defining information. The SEI core set consists of the following set of metrics: Size, (Size, progress, reuse), Staff-Hours (Effort, cost, resource allocations), Calendar dates (Schedule), and Counts of software problems and defects (Quality, readiness for delivery, improvement trends). The SEI helps organizations implement these metrics through technical guidance. At the annual SEI sponsored Software Engineering Conference there is normally a session that discusses experiences with the use of the core metrics set. The point of contact for this activity is:
Anita Carleton
Software Engineering Institute (SEI)
(412) 268-7718, adc@sei.cmu.edu.

Order-I

This project is based on the Air Force initiative, the NSDIR. The National Software Data and Information Repository (NSDIR) initiative originated in a workshop sponsored by the Office of the Secretary of the Air Force for Acquisition, specifically Communications, Computers, and Support Systems (SAF/AQK), which is headed by Mr. Lloyd Mosemann. As a result of recommendations from the workshop Mr. Mosemann initiated the Order I project, an effort to prototype a capability to demonstrate the capabilities that a data repository could provide. Order-I utilizes a number of existing data sets such as those of the SEL and NUWC in order to demonstrate the capability for "viewing" data. The contract, initiated under the Air Force's Computer Technology Transition program, produced a demonstrable prototype. The Order-I program is evolving and new project sites are being solicited to collect additional data in order to provide for the collection of data in the future. The point contact for this initiative is:
Lt Col Dan Romano
SAF/AQK
(703) 697-3108, dromano@aqpo.hq.af.mil.

STSC Metrics Starter Kit

The Software Technology Support Center (STSC), operated for the Air Force at Hill AFB Utah, provides a metrics' starter kit based on the Air Force Software Metrics Policy. This policy requires that software metrics cover cost, schedule, and quality. These areas are subdivided to capture the following "core" attributes: size, effort, schedule, software quality and rework. The STSC also offers software measurement courses to DoD sites. The point of contact for this activity is:
STSC Customer Service
(801) 777-8045, portr@software.hill.af.mil.
U.S. Army STEP Metrics Program

The U.S. Operational Test and Evaluation Command (OPTEC) initiated the Software Test and Evaluation Panel in 1989. One of the products of this panel activity was the development of a standard set of twelve metrics that has since been called the STEP metrics set. The metrics set is: Management: Cost, Schedule, Computer Resource Utilization, and Software developer maturity; Requirements: Requirements Traceability and Requirements Stability; Quality: Design Stability, Complexity, Fault Profile, Breadth of Testing, depth of Testing, and Reliability. The Army is implementing this system in conjunction with a number of tools and training courses. The concept is to have data collected according to the defined metrics for use in decision making in programs and eventually across programs. The decisions would be aided through the use of tools such as the Software Metrics Management Information System (SMMIS). An Army repository will be created to house all of the data. Additional tools would provide higher levels of management to gain access to and insight into a wide array of Army programs. The point of contact for the STEP program is:

Don Scott Lucero
OPTEC (703) 756-0895, lucero@optec.army.mil.

NASA Goddard Space Flight Centers Software Engineering Laboratory (SEL)

The SEL is based on the implementation of two quality improvement methodologies: the Experience Factory and the Quality Improvement Program (QIP). The Experience Factory is an organizational concept that provides an analysis and synthesis capability, as well as a "library" facility for data, products, and information about software development. It not only captures products and data, but the experiences of the management and engineering staff involved in software development, ergo the term "experience factory". The measurement program in the SEL is based upon the Goal/Question/Metrics paradigm, a method that aids in the definition, selection, and interpretation of measures of the software process and product and other forms of software experience. This measurement approach is used by the Quality Improvement Program to support continuous improvement based on six steps: Characterize, Set Goals, Choose Process, Execute, Analyze, and Package. It represents for development what the Shewart-Deming Cycle Plan/Do/Check/Act represents for quality management programs. The SEL has been performing experiments, developing systems, and collecting data since 1976 (Reference 1). In 1994 the SEL was awarded the SEI and IEEE Process Achievement Award for these activities. The point of contact for this program is

Rose Pajerski
SEL, (301) 286-3010, rose.pajerski@gsfc.nasa.gov.

Reference 1. Available from the DACS, the NASA/SEL dataset, ordering information from the DACS Order Form.
Rome Laboratory has been conducting a research and development program in requirements engineering since 1985. One result has been the evolutionary development of the Requirements Engineering Environment (REE). The REE is an integrated set of tools which allows system analysts to rapidly build functional and user interface prototypes. The REE is currently comprised of two tools: Proto and RIP.

Proto

Proto supports the prototyping of functional requirements. A functional prototype can model and execute the target system's operational capabilities to check the completeness and correctness of information as it flows through the system. This type of prototyping is typically performed on critical or complex applications. Proto supports the functional prototyping of requirements which are targeted to both sequential and parallel architecture environments. When performing functional prototyping the requirements analyst first obtains functional specifications for the application or the subset of the application to be prototyped. Such functional specifications represent precise descriptions of what the system does.

The analyst uses Proto to graphically construct a software model (logical model) which captures the functional specifications of the application. During this process the analyst creates multiple diagrams to identify the various functions in the specifications and their corresponding relationships. Such diagrams can be based on dataflow or object-based paradigms. The analyst uses a graphical editor to create all of the data objects required by the application. The details of the various specification functions are captured in a high level language which was explicitly developed for the tool. C functions and/or binary executables can also be used to describe function activities. In addition, a reusable library of functions is available to expedite model development. Collectively, the various diagrams, data objects, language statements and library functions make up a Proto software model.

If the software is targeted for a parallel architecture environment or the analyst wishes to identify and validate potential parallelism in the application being prototyped then a hardware model (physical model) can be graphically created and associated with the software model. The analyst uses a graphical approach to create a hardware model which consists of processors, buses and memories. In addition, Proto provides configurations for two multiple-instruction, multiple data (MIMD) parallel processing hardware architectures: a shared memory architecture and a distributed memory architecture. A template for automatic creation of mesh interconnection topologies is also provided. Once the hardware model is
defined, the logical software functions can be associated with the physical hardware components. More than one such software-to-hardware mapping can be associated with a given hardware model. Moreover, more than one hardware model can be associated with a given functional prototype. Such capabilities support the effective analysis of various software to hardware mappings and hardware architecture trade-offs.

The analyst then executes the functional prototype using the Proto interpreter facility. Functions which are currently executing in the software model are highlighted. More importantly, the analyst can set breakpoints and view various data values. Performance statistics are available after an interpretation session. Once the application has been debugged, it is executed for the end users to verify its completeness and correctness. Comments provided by the end users are incorporated into the requirements specifications and the functional prototype. The prototype is once again executed for the end users. The procedure of executing the prototype for the end users and incorporating comments into the requirements specifications and prototype continues until the functional requirements are validated. The REE provides the capability to generate Ada and C code from the functional specifications. Such code can serve as the starting point for further design work.

**Rapid Interface Prototyping (RIP)**

RIP provides a graphical toolset to prototype human-machine interfaces. Such interfaces can model the system's screen layout as well as execute its associated functions. One of the strengths of RIP is that its users need not be computer scientists or programmers. RIP employs an Apple Macintosh "McDraw-like" graphics editor to create the objects required for the prototype and menus to specify the activities which will occur during the prototype execution. There is also a world database facility which allows geographical areas to be extracted out of a database and incorporated into the prototype.

The requirements analyst and end-user work together to define and capture the user interface requirements. Once requirements are defined, the analyst identifies relevant graphic objects and constructs a set of static and dynamic displays to be used in the prototype. The graphic objects represent the real-world objects as the user would see them on the display. Using menus, the analyst then links these displays in a logical sequence to create event sets and scenarios which represent operational system requirements. Examples of functionality which can occur during the scenario are displaying objects, ringing alarms, flashing objects, and prompting the user for inputs. RIP also supports the creation of simulated real-time dynamic displays and object motion which are based on events occurring over time. Once completed, the analyst working with the user, can execute the RIP prototype by selecting predetermined "hot-buttoned" objects on the display. Typical comments that users might provide are the need to change the color of an object, relocate an object on the screen or build additional functionality into the human-machine interface.

**Proto and RIP Integration**

The REE provides data sharing between the Proto and RIP tools. The RIP and Proto tools are integrated such that there is two way data sharing between them. First, a Proto model can be used to drive a RIP user interface prototype. The Proto tool provides a set of predefined functions which facilitate the manipulation of RIP objects. Such manipulations include removing and displaying objects, changing the color of objects, setting data values and moving objects to new locations. No programming is required to connect a Proto model with a RIP user interface. The second type of data sharing between Proto and RIP allows data values specified in RIP to be read and manipulated by a Proto model. Consequently, the data
required for a Proto model can be interactively entered using custom made, user friendly interfaces. Once again, non-programming techniques are used to specify the connection between the two tools.

Utility & Usability

Proto is particularly useful for constructing system topological structures, logic control of software/system behavior and general modeling of the system's information/data flow triggered by events or external stimuli. RIP is useful for conceptualizing and designing user interfaces consisting of menus, windows and screens. In addition, its flexible graphical editing capabilities and other tools (such as map generation and scenario generation) make it a valuable resource in any organization performing software/system development activities. The integration capability provides the mechanism to execute the entire target system prototype very easily. Prototypes can be developed in a timely fashion and presented to key players of the lifecycle (end users, analysts, designers, developers) during appropriate system development milestones and reviews. REE prototype products can be thought of as low cost "mediums of communication" in which all system development players can observe prototype products and reconcile assumptions, tendencies, preferences and perspectives about requirements and high level design issues. At these reviews, feedback is then elicited and incorporated into the prototype and then reevaluated as necessary until all players are at a proper orientation point in which to proceed with more traditional development activities.

Operational Requirements

REE currently resides on a Sun platform (4/65 or better) with a minimum of 32 MB RAM, minimum of 100 MB of swap space and high resolution (color preferred) graphics monitor. Required software includes Sun OS 4.1.2 or 4.1.3 (porting to Solaris is currently underway), Ontologic's ONTOS 2.2 object-oriented database management system, Sun C compiler, Sun C++ 2.1 or 3.0.1 (if compiling generated C++ code) and Sun Ada (if compiling generated Ada code).

Rome Laboratory is actively transitioning REE technology to the general technical community. All inquiries can be directed to:

James L. Sidoran
Rome Laboratory
525 Brooks Road
Rome, NY 13441-4505
(315) 330-2762; DSN 587-2762
james@se.rl.af.mil.

Use these buttons to navigate between newsletter articles.
[Previous Article] [Next Article] [In This Issue]
Last year, in an attempt to focus this publication, the DACS formed an Editorial Board of distinguished professionals involved and concerned with measurement in software engineering practice. The Board meets four times a year, by teleconference. The objective of this activity is to seek outside advice and guidance on the form and content of the newsletter. The members of the Editorial Board, with the exception of the Kaman Sciences' personnel, all donate their time and we thank them for their interest and participation.

- Vic Basili, University of Maryland
- David N. Card, Software Productivity Solutions, Inc.
- Anita D. Carleton, Software Engineering Institute
- Andrew J. Chruscicki, Rome Laboratory
- Lon R. Dean, DACS Newsletter Editor
- John J. Marciniak, Kaman Sciences Corporation
- Jack McGarry, Naval Undersea Warfare Center
- Rose Pajerski, NASA Goddard Space Flight Center
- Nancy L. Sunderhaft, Kaman Sciences Corporation
- Dolores Wallace, National Institute of Standards and Technology

Use these buttons to navigate between newsletter articles.
[Previous Article] [Next Article] [In This Issue]
Please come visit our booth at the Software Technology Conference '95 and pick up our free information materials. You can talk with John Marciniak, author of "Encyclopedia of Software Engineering," and talk with James DeLude, our resident Web Developer.

Use these buttons to navigate between newsletter articles.
[Previous Article] [Next Article] [In This Issue]
Under contract to the Program Executive Office for Cruise Missiles and Unmanned Aerial Vehicles (PEO-CU), Science Applications International Corporation (SAIC) has developed a set of criteria to facilitate the management of system development milestones (technical reviews and audits such as Critical Design Review and Functional Configuration Audit) for a system development effort. These milestones are usually meetings that require a significant amount of preparation and travel. The presence of many individuals such as users, acquisition managers, systems engineers, integrators, developers, test engineers, and regulators is required. Much time, effort and money may be wasted for a milestone that has not achieved a state of readiness that assures successful completion.

A management tool to aid in determining both readiness and successful completion is very useful. The criteria provides a means to verify that the stage is adequately set for a milestone to begin and that the milestone adequately complies with requirements and standards. The criteria are provided in the form of a checklist applicable for each milestone. The checklist enumerates items that are the responsibility of various players of the system acquisition process (e.g., the system acquirer, the system developer, the Independent Verification and Validation agent). This checklist is intended to identify, if appropriate, products and activities that are not complete, available, or of sufficient quality, and provides a basis from which to enforce standard practices. The likelihood of successful completion of an event and the most judicious use of resources is therefore increased. The framework for the criteria is based on IEEE STD-1074 (Standard for Developing Software Life Cycle Processes) and the requirements of MIL-STD-1521B. For each system development milestone, a set of inputs, entry criteria, activities, outputs, exit criteria, and metrics are listed and a point-of-contact responsible for each listed item is declared. Each set is packaged into a "kernel" (see Figure 1). There is one "kernel" for each specific milestone event. The complete set of kernel elements provides a basis for determining the successful completion and pass/fail of a system development milestone event. Deviations from items listed in the kernels are risk items that should be mitigated.
Entry criteria are the conditions that must be present before a milestone event can occur. Inputs are the products (documents, briefing material, agreements, etc.) that must be available and the activities that must be completed in preparation for the event. Strict adherence to the entry criteria and inputs provides a high level of assurance that appropriate products are available and have been verified to be of sufficient quality to begin the milestone event. Actions are the activities that occur during the course of the event. Outputs are the products that must be available for the event to be considered complete. Exit criteria are the conditions under which the event is complete. Metrics are based on a basic set of metrics mandated for the management of DoD systems acquisition in such references as DODI 5000.2 and ASAF Acquisition Policy 93M-17. Specific size, effort, schedule, quality, and rework metrics are defined for each milestone. Kernels are available for the following system development milestone events: System Requirements Review (SRR); System Design Review (SDR); Preliminary Design Review (PDR); Critical Design Review (CDR); Integration Readiness Review (IRR); Test Readiness Review (TRR); Physical Configuration Audit (PCA); and Functional Configuration Audit (FCA). For more information contact:

Lillian K. Zelinski  
Science Applications International Corporation  
1213 Jefferson Davis Highway  
Suite 1500  
Arlington, VA 22202  
(703) 414-3940; FAX (703) 414-8267  
zelinski@cg4.saic.com
Office Worker Productivity

The Delphi Consulting Group, Inc., (Boston, Massachusetts), a leading electronic document management system and workflow management information system consultant, points to disturbing research in office worker productivity (Reference 1). While information technologies have led to double digit productivity gains on the factory floor, net white collar productivity over the last 20 years has increased less than one percent! While it is true that white collar workers have increased total output, the increase in proportion to expenditures on new office information systems technology has resulted in almost no net gain. This research indicates that information technology has failed to deliver on the promise of office productivity gains. Add to this, noted MIT economist, Lestor Thurow's finding that the industrial global complex is now able to produce 130% of the hardgoods the world can consume (Reference 2). Does this mean that had we succeeded in increasing office productivity, we still would have failed because the world could not absorb the increase? What is the measure for productivity success?

It appears the "more-is-better" measure of productivity is not the best barometer of productivity gains. In the 1970s and 1980s Japanese automobile manufactures did not outsell U.S. automobile manufactures because of their ability to produce more cars. They reacted to the changes in the marketplace, namely the demand for better quality. If productivity and success are measured by the ability to manage change, shouldn't information processing technologies be measured by their ability to help organizations adapt to change?

Managing Change

The basic elements of a business process include people, money, and time. Developing an equation that makes the best use of these basic elements in relation to the market is, and has always been, management's challenge. The 1990s have ushered in a business process renaissance. Driven by world economic events, organizations are redefining and redesigning the way they do business as a means of staying competitive. In the DoD, this means adopting new business models that allow the U.S. to remain militarily superior given a new world order, and its associated impact on the defense budget. Business Process Reengineering (BPR) and Total Quality Management (TQM) are recognized terms that embody this renaissance. Essentially, TQM and BPR are business models that seek to improve service or product
quality while simultaneously eliminating or restructuring redundancies and inefficiencies. In effect, TQM and BPR are models that help manage change. Public and private business organizations are employing TQM and BPR models, and because we live in a information society, are implementing Workflow Management Information Systems to assist them. Workflow Management Information Systems are a relatively recent development that evolved from Electronic Document Management Systems (EDMS). Business processes include an exchange of information. Increasingly, this information is in an electronic format. Electronic Document Management Systems allow electronic information in a variety of formats (i.e., documents, forms, databases, drawings, images, video files) to be received, stored, retrieved, and transmitted. Workflow Management Information Systems provide a mechanism to effectively integrate electronic information in the business process and, more importantly, allow for dynamic changes in the information flow.

**Workflow Management and the DoD**

No other market sector is witnessing more dramatic business process changes than the DoD, particularly in the area of technology acquisition. The mission to be better prepared and technically superior to the former Soviet Union was clearly understood by DoD military and civilian personnel and the contractors who supplied them with products and services. Business models were established for weapons procurement that dictated superior performance first with cost as a dependent variable.

The new business model, recently advanced by Paul Kaminski (Reference 3), Under Secretary of Defense, Acquisition and Technology continues to reflect a strategy of technical superiority, within an affordable budget. This strategy will be implemented by smaller deployable forces and leaner defense support organizations, reduced life cycle costs of weapons systems, acquisition reform, greater reliance of the commercial industrial base, and leveraging our allies' industrial base.

Implementation of the new acquisition business models advanced by Paul Kaminski will be performed by leaner organizations. Leaner organizations are in order due to a sharp reduction in production requirements. The individual tasks required to complete the weapons acquisition process however, most likely will not reflect a corresponding reduction. In fact, at least in the short run, increasingly complex weapons systems, environmental effects, and changing procurement standards may actually increase the number of tasks required to complete an acquisition process. Completing the business acquisition process more efficiently with leaner organizations will require flexible information systems that can adapt to rapid changes in the business process. Information systems that support reduced "time-to-market" and electronic information exchange with the commercial industry will be valued. Workflow Management Information Systems are well suited to these requirements.

Workflow Management Information Systems integrate electronic information to support business processes. Numerous information tools (computer-aided design, imaging, forms processing, databases, spreadsheets, word processors, etc.) support individual tasks, but too often these tools work in isolation from one another. Workflow Management Information Systems connect these discreet information tools in a series of tightly coordinated parallel processes. Information flows in a manner that supports the most efficient method for completing the business process.

**DACS Workflow Management Information Systems**

Interestingly, the DACS has designed, developed, and successfully implemented Workflow Management Systems long before the term "workflow" was coined. These systems have been literally saving the DoD
millions of dollars in the form of time-to-market and labor costs.

The DACS has designed a series of reusable codes that provide users with a workflow development environment. These codes have been packaged and released under the name "OASYS," Open Architecture System. The name is revealing in that the DACS designed these series of codes to support open system standards. Standards such as X-Windows, SQL-compliant databases, "C" and "C++" languages are included within the OASYS design. OASYS includes system administration, forms development, dynamic document routing, auditing, and database integration components.

The DACS has employed OASYS to provide workflow automation for DoD acquisition and technology procurement functions. The DACS has implemented a Procurement Package tracker for the U.S. Army at the Picatinny Arsenal in Dover, New Jersey. This system automates the complex engineering process of weapon certification in preparation for new vendor solicitation requests. The Procurement Package Tracker has been credited with helping the Picatinny Arsenal reduce the time it takes to process Technical Data Packages (TDPs) from an average of 198 days in 1988 to less than 30 days today. During this same period, the Arsenal's TDP error rate dropped from 33 percent to less than 1 percent! At the U.S. Army Watervliet Arsenal, OASYS was used to develop an Automated Acquisition System that routes procurement requests through fund certification, hazardous materials, and concurrence processes. The Defense Technical Information Center (DTIC) is using OASYS to process Technical Area Tasks (TAT) for Information Analysis Center (IAC) Programs. DTIC's TAT Tracker functions include electronically generating Statements of Work, Work Plans, Technical and Cost Proposals, and other supporting documentation, and routing this information through the DoD for required reviews and approvals. The TAT Tracker is expected to drastically reduce the time from a technical task initiation request to task commencement.

"Close Up Corner" in the next DACS Newsletter will take a closer look at the Procurement Package Tracker. We will discuss in more detail the components of the Procurement Package Tracker and how this system is helping the U.S. Army adjust to business process changes.


The 19th Annual Software Engineering Workshop was held at NASA Goddard Space Flight Center on November 30 and December 1, 1994. The workshop is sponsored by the Software Engineering Laboratory (a joint venture of NASA/Goddard, Computer Sciences Corporation, and the University of Maryland). Following SEL Director Rose Pajerski's opening remarks, SEL personnel reported on recent research in the Software Engineering Laboratory, addressing work underway there in experience-domain analysis, maintenance process characterization, and the experience "packaging" process.

Vic Basili (University of Maryland), explored the possibilities of expanding the SEL's quality improvement paradigm by examining ways in which "local" experience, a hallmark of the SEL improvement concept, might be transferred beyond the local environment to other software organizations whose product and process may be similar to benefit from that experience. He proposed both quantitative and qualitative approaches for assessing the applicability of experience across domains (or organizations). Basili's structure for comparing software experience domains proposed a scientific method for globalizing local software engineering experience.

Jon Valett (NASA/Goddard), the new Director of the SEL, reported on maintenance research. Valett's study examined software in the SEL's domain, where some of the operational systems are more than 10 years old. To baseline, or develop an accurate understanding of the local maintenance process, the research team looked at organizational structure, process activities, and product attributes. Valett itemized some key differences to keep in mind when studying an evolving product: look at "release" characteristics, measure effort per software change, distinguish between operational errors (those found in the operational product) and "release" errors (those introduced during maintenance), and apply experience gained from this product to improve future releases of the same product.

In the third presentation of the session, Sharon Waligora (Computer Sciences Corporation) explained how the SEL applies its lessons learned and improved approaches in the development environment. Experience "packaging," she said, is the process of culling the most relevant data and findings from research, experimentation, and process improvement activities; synthesizing them into useful information and guidance; and returning them to the source, the projects that supplied the raw experience. The SEL packaging process, Waligora said, can be viewed as a two-way feedback loop between the development organization and the experience "packagers." In one direction, the loop takes in the experience and expertise of development/maintenance personnel and identifies their needs. That qualitative input, as well as quantitative data and results of experimentation with new technology, is then synthesized by professional packagers. Waligora shared some data and lessons learned from SEL packaging efforts.
Cost for packaging varied according to the format and level of detail of the material, averaging around 1 to 1.5 percent of the software development budget. In addition to the SEL presentations, speakers from academia, industry, and other NASA organizations discussed process models, software certification, measurement, and other topics.

Daniel Roy (Software Technology, Process & People), who recently completed a term as a visiting scientist at the SEI, reported on his experience there using the Personal Software Process (PSP). The PSP is a software engineering approach intended to illuminate for the individual the importance of good software engineering practices. Studying the PSP, Roy completed several programming exercises that incorporated measurement activities, such as error logs and effort logs, project planning and tracking, and post-mortem reviews. According to Roy's experience, the PSP does bring software engineering practices into focus, highlighting for the individual the connection between performing these activities and improving one's own software product.

Stan Rifkin (Master Systems Inc.) developed training in program comprehension techniques. Rifkin pointed out that reviewer comprehension is a factor often overlooked in inspections because it is assumed (wrongly) that developers, in their role as inspectors, know exactly how to go about reviewing software artifacts. Rifkin described program comprehension as a "complex process of translating, interpreting, and hypothesis testing among domains—from the real world, to the application domain, to the computing domain." Rifkin deployed his training and inspection technique in a commercial setting, where he trained a group of 35 software developers already using software inspections as a part of their process. He then followed up by collecting data over two years to quantify the effect of the training on their customer-detected-defect rate. Two control groups were also tracked. While reported errors for the two control groups remained stable at their initial levels, Rifkin reports a steady decrease in number of customer-detected defects in the products delivered by his group. This drop leveled off at 90 percent fewer customer-detected defects per day.

Scaling-up measurement to a broad population, Craig Sinclair (SAIC) applied the SEL baselining technique across NASA. The research was sponsored by the NASA Software Engineering Program. The specific goals of the baseline study were to provide insight to NASA management on the scope and nature of its software; to help focus and direct future spending on training, technology, and software engineering; and to identify areas ripe for software product and process improvement. Sinclair's team learned that more than 10 percent of NASA personnel and support contractors work with software the majority of their time on the job. One billion of NASA's 14-billion-dollar budget goes to building software and other software-related costs. This software can be divided into several domains, the two largest of which are mission ground support (35 percent) and administrative (26 percent) software. Programming language preferences and usage trends (always an intriguing bellwether) were examined. Far and away, most new systems are being built using C/C++ (47 percent), with FORTRAN a distant 29 percent, and Ada trailing with a distant 11 percent for new development projects. This is in contrast to operational systems, which are dominated by "other" languages (e.g., Assembler, Jovial, Lisp, Pascal) (38 percent), followed by FORTRAN, C/C++, COBOL, and Ada. Understanding and assessing NASA products and processes at this level is an important first step toward developing Agency policy and guidance for local NASA organizations at the Center level and below.

The SEL is accessible on the World Wide Web:

The workshop proceedings will be available in April free of charge from:
Software Engineering Laboratory
Software Engineering Branch
Code 552
NASA Goddard Space Flight Center
Greenbelt, MD 20771.

[Previous Article] [Next Article] [In This Issue]
7th Annual Software Technology Conference
"Architecting the Information Highway for the Warrior"
9-14 April 1995
Salt Lake City, Utah
Dana Dovenbarger
Lynne Wade
Software Technology Support Center
Ogden-ALC/TISEC
7278 Fourth Street
Hill AFB, UT 84056-5208
(801) 777-7411, FAX (801) 777-8069
DSN 458-7411, DSN FAX 458-8069
dovenbar@oodis1.hill.af.mil
lynwade@oodis1.hill.af.mil

17th International Conference on Software Engineering (ICSE)
24-28 April 1995
Seattle, Washington
Dewayne Perry
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974
(908) 582-2529, FAX (908) 582-7550
dep@research.att.com

IEEE Real-Time Technology and Application Symposium
15-17 May 1995
Chicago, Illinois
Ted Baker
Department of Computer Science
Florida State University
Tallahassee, FL 32306-4019
(904) 644-5452
baker@cs.fsu.edu

5th International Conference
on Software Quality (5ICSQ)
23-26 October 1995
Austin, Texas
Herb Krasner
Krasner Consulting
1901 Ringtail Ridge
Austin, TX 78746
(512) 328-4264, FAX (512) 328-3260
hkrasner@cs.utexas.edu

Check out the DACS Calendar of Events for other upcoming Software Engineering Events.

Use these buttons to navigate between newsletter articles.
[Previous Article] [Next Article] [In This Issue]
You can access these reports through our [World Wide Web Site](http://www.dacs.dtic.mil). You can view the reports by clicking on the technical reports hyperlink on the first page of the DACS Home Page. Documents with an (abs) after the name are abstracts, we have not put these documents on-line. The following DACS technical reports are currently available:

- Electronic Publishing on the World Wide Web: An Engineering Approach
- A Study of Software Management: The State of Practice in the United States and Japan,
- Artificial Neural Networks Technology,
- A Review of Non-Ada to Ada Conversion, and
- Software Design Methods.

If you would like a bound copy you can [order them](mailto:dacs@dacs.dtic.mil) from the DACS for a nominal fee. The bound copy prices and ordering information are listed on page 11 of this newsletter. The internet addresses for the DACS are listed below:

WWW URL: [http://www.dacs.dtic.mil](http://www.dacs.dtic.mil)
Anonymous FTP: [ftp.rome.ittssc.com](ftp.rome.ittssc.com)
Discussion Groups: [listserv@dacs.dtic.mil](mailto:listserv@dacs.dtic.mil)
E-mail: [dacs@dacs.dtic.mil](mailto:dacs@dacs.dtic.mil)

Use these buttons to navigate between newsletter articles.
[Previous Article] [Next Article] [In This Issue]
The DACS Newsletter is the current awareness publication of the Data & Analysis Center for Software (DACS), a centralized source for current, readily available data and information concerning software engineering and measurement technology. The DACS is a Department of Defense (DoD) Information Analysis Center (IAC) which is administratively managed by the Defense Technical Information Center (DTIC) under the DoD IAC Program. The DACS is technically managed by Rome Laboratory (RL) and operated by ITT Systems Corporation.

Suggestions and submissions of articles for inclusion in the DACS Newsletter are welcome and encouraged. Inquiries and comments regarding DACS capabilities, products, or services may be addressed to our DACS Newsletter Editor (news-editor@dacs.dtic.mil). The DACS can be contacted directly by phone at (315) 334-4905 or by FAX at (315) 334-4964.
The 10th Knowledge-Based Software Engineering Conference (KBSE-95)
Call for Papers, Demos, Panels, and Tutorials
November 12 - 15, 1995
Boston, Massachusetts, USA

Call for Papers
The Knowledge-Based Software Engineering Conference provides a forum for researchers and practitioners to discuss applications of automated reasoning, knowledge representation and artificial intelligence techniques to software engineering problems. This conference focuses on specific knowledge-based techniques for constructing, representing, reasoning with, and understanding software artifacts and processes. These techniques may be fully automatic, may support, or cooperate with humans.

KBSE-95 encourages contributions describing basic research, novel applications, and experience reports. Recent successful contribution topics include: applications, automating software design and synthesis, education, maintenance and evolution, process management, program understanding, requirements, reuse, security, user interfaces and human interaction, as well as validation and verification.

In addition to technical papers, KBSE-95 includes tutorials, panel discussions, and project demonstrations. The location of this year's conference is in Boston, Massachusetts.

Paper submission timetable (subject to revision):
Paper submission May 1, 1995
Notification Mid June 1995
Camera-ready copy July 1995

Papers should not exceed 6000 words in length, with full page figures counting as 300 words. Papers will be reviewed by at least three members of the program committee using the following criteria: technical quality, originality, clarity, appropriateness to the conference community, and adequacy of references to related work. Six copies of the paper submission should be sent to the Program Chair.
Call for Panels

KBSE has a history of stimulating and thought-provoking panel discussions. If there is a KBSE-related topic that you would like to see a panel devoted to, and you are willing to organize this panel, you are strongly encouraged to submit a panel proposal. Panel proposals should consist of roughly two paragraphs describing the basic issue to be discussed and why you think it would be well suited to a panel format. A paragraph outlining each viewpoint that each panel member represents, and a tentative list of panelists should also be included. The proposal should contain the E-mail address and phone number of a single contact, and should be sent to the Tutorial/Panel Chair by May 1, 1995. Electronic submissions are preferred.

Call for Tutorials

The KBSE conference represents many diverse and highly specialized fields, and we encourage experts in any of these to take advantage of the opportunity to present their material in full or half day sessions. Tutorials will be the day before the conference, and have been effective in bringing newcomers up to speed for understanding technical discussion during the conference and in educating participants in the details of particular projects and technologies. Tutorial proposals should consist of a description of the topic, outline if available, a bibliography, and the qualifications of the presenter(s). Specify the target audience and any previous knowledge/experience required, and the tutorial goals in terms of the benefit to participants. You should estimate the materials tutorial attendees will require and the cost of preparation. Please specify whether you would prefer a half day or full day format. The proposal should contain the E-mail address and phone number of a single contact, and should be sent to the Tutorial/Panel Chair by May 1, 1995. Electronic submissions are preferred.

Call for Demonstrations

The conference will also host software demonstrations. Research prototypes and commercial products with some knowledge-based aspect are welcome. Please contact the Demo Chair by June 1st, 1995.

Program Chair:
Dorothy Setliff
341 Benedum Hall
Department of Electrical Engineering
University of Pittsburgh
Pittsburgh PA 15261
setliff@ee.pitt.edu
FAX: (412) 624-8003

General Chair:
Howard Reubenstein
GTE Laboratories
40 Sylvan Road
Waltham MA 02254
hbr@gte.com
Tutorial/Panel Chair:
Chris Welty
Computer Science Dept.
Vassar College
Poughkeepsie NY 12601
welty@cs.vassar.edu

Demonstration Chair:
Richard Piazza
The MITRE Corporation
Mail Stop K331
202 Burlington Rd.
Bedford, MA 01730
rich@mitre.org

Steering Committee:
Robert Balzer (USC/ISI), Barry Boehm (USC), Dave Fisher (NIST), Cordell Green (Kestrel Institute.),
Mehdi Harandi (University of Illinois), Lewis Johnson (USC/ISI), Peter Selfridge (AT&T), Mike Evangelist (Florida International University), Doug White (Rome Laboratory).

Program Committee:
Paul Bailes (University of Queensland), Paul Bailor (AFIT), Kevin Benner (Andersen Consulting),
Wolfgang Bibel (Technische Hochschule Darmstadt), Alan Bundy (University of Edinburgh), Joe Carozzoni (Rome Laboratory), Paul Clements (SEI), Martin Feather (USC/ISI), Steve Fickas (University of Oregon),Mehdi Harandi (University of Illinois), Karen Huff (GTE Labs), Richard Jullig (Kestrel),
Lewis Johnson (USC/ISI), Gail Kaiser (Columbia University), Axel van Lamsweerde (Universite
Catholique de Louvain), Mike Lowry (NASA Ames), Gordon Novak (University of Texas), Walt
Scacchi (USC), Peter Selfridge (ATT), Doug Smith (Kestrel), Loren Terveen (ATT), Richard Waldinger
(SRI), Chris Welty (Vassar), Doug White (Rome Laboratory), Alan Whitehurst (US Army CERL).

Sponsored by: Rome Laboratory
In cooperation with IEEE
Pending cooperation with ACM, AAAI

Administrative support by the Data & Analysis Center for Software (DACS), a Department of Defense
(DoD) Information Analysis Center (IAC) which is administratively managed by the Department of
Defense Technical Information Center (DTIC) under the DoD IAC Program. The DACS is technically
managed by Rome Laboratory (RL) and operated by Kaman Sciences Corporation. General Information:

Further conference information can be obtained by sending an electronic mail message to
kbse-info@cs.rpi.edu, through the World-Wide Web at URL:
http://sigart.acm.org/Conferences/kbse

or by writing to:
Data and Analysis Center for Software
PO Box 1400
Rome, New York
(315) 334-4905; FAX (315) 334-4964
cust-liasn@dacs.dtic.mil

You can also access this information and other conference information on the DACS Home Page.

Use these buttons to navigate between newsletter articles.
[Previous Article] [In This Issue]