"Ada-WOW"

Ada's Window On the World

SIGAda 2001

Sponsored by ACM SIGAda

Thunderbird Hotel
Bloomington, Minnesota, USA
2001 September 30 – October 4

Thursday, October 4
Volume 3, Issue 3

What O Week

by Ann S. Brandon (Onyons, Inc.)
SIGAda 2001 Ada-WOW Editor-in-Chief

WOW now stands for What O Week! I haven't been in a newsroom working as editor for over twenty years and I'd forgotten how many words need to be written, edited, and seduced out of people.

I want again to thank Steve Deller, who today writes about Bill Pritchett's talk on OO metrics for Ada95. Clyde Roby earns a shining star again by covering two sessions, Janusz Laski's talk on ASIS, and Dhavy Gantsou's discussion on targeting Ada 95 for communications networks. Dave Cook edited me (getting revenge for my editing him yesterday), and Currie Colket dedicated a late night to writing his overview of the conference.

The newsletter is stuffed with articles on yesterday's sessions, including a long one I wrote weaving together Martin Carlisle's morning keynote speech and the various lessons learned session on teaching from presenters, tutorials, and SIGAda educators. I assume you'll have read enough of me after that and the article on Tucker Taft's keynote address, so I'll keep this short.

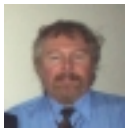
Of course, without Hal Hart there would be no announcement of the Awards Ceremonies. In fact, there would be no awards, since he was the one who badgered us all into nominating these great guys. There would be no Ada-WOW, no photos, not impetus, no guilt over sleeping more than three hours. He's the center of the newsletter, the book between the ends, the O that holds together the Ws.

Thank you for all your terrific participation, and I hope to see you next year.

Award Winners Alan Burns, Clyde Roby, and Currie Colket

SIGAda Awards

by Hal Hart (TRW),
SIGAda Awards Comm. Co-Chair



On Wednesday SIGAda awards for ***Outstanding Ada Community Contributions*** were awarded to

- **Alan Burns** *(Univ. of York),* for pioneering contributions in the area of realtime features, co-authoring the definitive work "Concurrency in Ada," and organization of the bi-annual Real-Time Ada workshop,
- **Charles McKay** *(Univ. of Houston/Clear Lake, Dir. of NASA/JSC-UHCL Software Engineering Research Center)* for long-time worldwide Ada advocacy, pioneering Ada runtime environment contributions, and principal responsibility for Ada in the International Space Station, and
- **Alfred Strohmeier** *(Swiss Federal Institute of Technology)* for tireless promotion of object orientation and Ada, long-time service as a Distinguished Reviewer for the Ada revision, ASIS and ASIS for GNAT development leadership, and major contribution to the revival of the Ada-Europe conference series.

SIGAda Distinguished Service Awards went to

- **Currie Colket** *(Mitre Corp. and current SIGAda Chair),* for long-time Chairmanship of SIGAda's ASIS Working Group, four years as SIGAda's Vice Chair for Meetings and Conferences, and numerous other SIGAda and Ada activities such as Navy Deputy AJPO Director dating back into the 1980's, and
- **Clyde Roby** *(Institute for Defense Analysis and current SIGAda Secretary),* for long-time behind-the-scenes efforts such as SIGAda conference proceedings editor and webmastering &

Inside this Issue

Weather Forecast

Today (Partly Cloudy)
High: 23 C (61 F)
Low: 5 C (near 40 F)

Friday forecast: ☺
High: 100 F

Weekend forecast: ☺
Low: 29 F

scribing for the ASIS Working Group, as well as similar roles in nationally important Ada activities dating back to the mid 1980's.

Read more about SIGAda Award winners in the upcoming issue of Ada LETTERS.

Congratulations to all winners, and thanks for your contributions!

Retiring Officers S. Ron Oliver, Bard Crawford, and Hal Hart receive miniature Ada statuettes as recognition for their service.

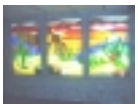
Tucker's Language Philosophy

by Ann Brandon (Onyons, Inc.)

S. Tucker Taft of AverCom, Inc., a Titan Corporation, began his keynote address by confessing that, as an undergraduate, he believed that software engineers' research and creative thought could result in their designing a "perfect language." With corporate and academic experience, however, he has realized that, though a perfect language might exist, only if deep pockets funded it and academicians wrote books about it would the language become popular. "Perception is reality in this world," he said.

Experience has not dampened his idealistic spirit. Tuck's keynote speech outlined how a language's merit can be measured in large part by how well it catches errors in compile- rather than runtime. "I believe in finding errors in the beginning"

Tuck is heading up the Ada0X effort, and thus his musings on how Ada can improve its compiletime checking are not merely philosophical. His belief is, "Careful annotation language can produce a highly flexible and usable language that nevertheless has no language-defined runtime failure conditions." When shown on a viewgraph to a room full of Ada programmers and tool vendors who are already having a hard time converting clients and managers to Ada95, however, it is more a gauntlet thrown than a theory aired. The Ada community gave Tuck a lively debate after his talk about some of his suggestions' ramifications from both users' and tool vendors' perspective. No doubt the debate will not end until Year X.



Education Wrap-up

by Ann Brandon (Onyons, Inc.)

Four educators prominent at SIGAda 2001 offered their lessons learned on the ease of teaching Ada at the college level and the difficulty of introducing an Ada-based course to the curriculum. Some of the basic questions that they answered concerned how to attract students to and then retain them in Ada courses.

Ada at the Air Force Academy

Martin Carlisle, associate professor at the US Air Force Academy in Colorado Springs, Colo., opened Wednesday's session with a keynote speech on his experience teaching Ada in core and elective courses. As an Ada teacher, he has the advantage of a "mandate," and sees the 200 computer science majors all through their academy training.

Martin himself prefers to teach Ada because of its clear syntax and software principles. But fundamentally Ada is what he calls a "cheap" language for students.

In order to graduate, the academy cadets need to satisfy 154 credits, whereas 124 are the norm in most universities. That's the time *demand* side. The time *supply* side is equally constrained, as they have no academic summer semester. Instead, they fulfill military duties during the summer. As a result the cadets majoring in computer science take a minimum of six and usually eight courses a semester, or two semester's worth of work in any civilian university.

The cadets' time crunch makes Ada the only language Martin can see as cost effective for training computer science majors. A move to the more popular languages would, in Martin's opinion, increase the cadets' learning curve. Furthermore, "We can't waste time tracking errors in C."

Martin himself is a convert to Ada, having completed his undergraduate and graduate work in C. "I would be hard pressed to say there are no bugs in my graduate dissertation," he admitted, "and certainly no critical ones."

Martin supported his claim of Ada's being "cheaper" in time and effort from a variety of sources. Rational, for example, discovered that costs for developing software in Ada were lower despite their having no experienced Ada programmers. Don Reifer's study showed a 50 percent error ration and lower cost. TRW demonstrated a big advantage in Ada in C^3 systems, especially in maintenance costs. The FAA in a Carnegie-Mellon Software Engineering Institute study comparing languages gave Ada the highest score of 78.8 out of 100. C++ scored 63.9, and C 59.5.

Finally, Martin mentioned John McCormick's study of the ACM's *Communications* article, "Hairiest Bug Stories," which described bugs that took several staff years to fix. McCormick discovered that Ada would have detected 88 percent of the programmer errors either during compile- or run-time.

"Boy Toys" in Teaching Ada

Martin also gave a presentation for Barry Fagin (another associate professor at the U.S. Air Force Academy) on their work in programming Lego robots in Ada using Ada/Mindstorms, which is available on the Web. Using toys to teach sequential programming control and event control inspires the cadets to give just that much more. He hopes to create a graphical simulator of the Lego's bumping, shaking, and wiggling, so cadets can program without competing for the lab's robots.

"Boy toys" have also helped SIGAda 2001 Program Chair John McCormick's annual Realtime with Ada course remain popular. In his advanced course, students write software to control model trains on a complex landscape of tracks. He teaches at University of Northern Iowa, Cedar Falls, where other languages reign. "It's easy showing Ada is appropriate for realtime applications," he admitted. "I would have a hard time convincing my department to use Ada for writing compilers."

Teaching Compiler Courses

John was referring to Tucker Taft's Wednesday talk on teaching a compiler course at Tufts University. While he considers writing a compiler the best single exercise for immersing a student into the entire world of software, he realizes that compiler courses are no longer popular to take or teach.

While Tuck encourages his students to program in Ada, he has never converted anyone. He went on to describe the language's attributes that he believes are custom-made for compiler creation.

A Popular Ada Course at A&M

Professor Salih Yurttas had presented a tutorial to an embarrassingly small audience on Monday. He talked Wednesday about his own course at Texas A&M University, Programming in Ada, which has attracted at least 35 students every fall since 1989. Though he uses no bells or whistles or "boy toys" to attract the students, he manages to retain them and make them happy; they neither drop the elective nor complain about it. As one of the two attendees of his tutorial, I can attest to his energy and enthusiasm, which he admitted have contributed to his reputation as an excellent teacher. Word-of-mouth have proven enough to continue filling his course.

Or is it Ada? According to John, at University of Northern Iowa's Freshman Sequence (CS 1 and 2), Ada lowered the usually high dropout rate. "In comparing the three or four years of C++ with the first two years of Ada, we retained 20 percent more students than with C++," he said. "Better yet, we retained 25 to 30 percent more women students."

DaimlerChrysler's Ada Cross Referencer Tool

Dynamic Analysis for Locating Product Features in Ada Code

by Mark Glewwe (Goodrich)

Laura J. White is a retired Naval Surface Warfare Officer. Today she lectures full-time at the University of West Florida. As a sideline, she paints expressionistic landscapes, which are shown at the Blue Morning Gallery in Pensacola,Fla.

Laura has brought Ada to life at the school by offering to teach it. As part of this effort she has joined forces with Norman Wilde to extend their existing Software Reconnaissance method to work with Ada code.

Software Reconnaissance is a public domain method developed at West Florida to work on C/C++ code. The language characteristics of Ada have provided new and interesting issues in the areas of embedded systems, multi-tasking systems, and realtime constraints.

The Ada adaptation started about five years ago and is being developed as part of the graduate student Capstone courses. The method consists of 1) Instrumenting the code, 2) Compiling the instrumented code, 3) Executing the code on a set of test cases, 4) Tracing the instrumented code, and 5) Analyzing the traces.

The primary result of the method is to identify several starting points to look for code that impacts the feature. The method offers an excellent tool to those analyzing either their own or someone else's code. It is especially useful when the code is completely unfamiliar

.The toolkit that supports the method includes an Instrumentor, a Trace Manager Interface/Trace Array Pakage, a Trace Manager, and TraceGraph, The method and toolkit are available in the public domain for download at *http://www.cs.uwf.edu/~recon/*.

This is an on-going development activity, so we should anticipate further progress in the future. Thank you Laura for sharing your experiences and insight.

Quote of the Day

In response to a failure in an Emergency Services System, a spokesman said:

"We are still not sure what the originating cause is, but it was definitely a software glitch and nothing more sinister." *-courtesy Peter Amey*

Beyond ASIS: Program Ddatabases and Tool-Oriented Queries

by Clyde Roby (IDA)

Janusz Laski's task all started with SWAT: SoftWare Analysis and Testing--a set of tools developed for the analysis and testing of software, now applied to Ada. It was originally written in Pascal for the analysis and testing of Pascal programs and called STAD: System for Testing and Debugging. SWAT objectives included an integrated approach to the analysis and testing of Ada programs.

The Laski team first tried to develop their own Ada parser, but then discovered ASIS, the Ada Semantic Interface Specification. ASIS was used to build the basic databases and the program databases using both a Static Analysis System (SAS) and a Dynamic Analysis System (DAS).

The program databases produced by SWAT are actually flat text files whose format was designed to be readable by both people and the programs of the system. The database files (DBs) are divided into system databases and procedure-level databases. System-level DBs represent the highest-level views (in general) of the program under analysis while procedure-level DBs contain information about individual subprograms. These are all fully described in the paper beginning on page 81 of the Proceedings.

During the development of SWAT using ASIS, the Laski team developed a secondary level of interfaces on top of ASIS which SWAT actually uses. Although they found ASIS extremely helpful, they believe that tool builders would be better served if they had a system of queries that does not need the knowledge of ASIS. This is particularly important since most of their research into the SWAT is language independent. Towards that goal, they proposed two higher-level sets of queries, as a generalization of program databases used in SWAT: program browser supporting queries and queries that support semantic program analysis. The latter should

allow tool builders to insert annotations either into the code or (preferably) into the flow graphs of the selected routines.

They also make an important claim: Dynamic Analysis should be an extension of Static Analysis, rather than an activity in and of itself. To support Dynamic Analysis, the tool builder should be able to do at least the following: First, it should be easy to instrument user-selected subprograms and control points within them with calls to a (tool-builder) execution monitor. Second, at each such call, the monitor should have full access to all the variables of the program, thus being able to retrieve their values. Third, another "active" query should also be available to set breakpoints in the selected routine and allow a change in the state of the program under analysis by redefining the values of selected variables before resuming the execution. Fourth, one should be able to dynamically evaluate the tool-user defined assertions (e.g., pre- and post-conditions, loop invariants). The instrumented code then would produce several Dynamic databases containing information about particular executions; these can then be used by specialized execution-based tools such as test coverage monitors, debuggers, etc.

Additional information is available at the SofTools website at www.softools.org.

Towards Predicting Quality with Metrics

by Steven Deller (Smooth Sailing LLC)

Bill Pritchett of DCS Corporation presented on Wednesday initial results from analysis of Ada95 programs using object-oriented metrics.

Functional metrics have long been used to predict the effort needed for software changes, track development to schedule and budget, and predict final code quality. With the introduction of OO languages, the challenge has been to define appropriate OO-based metrics.

A number of "pure" OO metrics has been proposed but there have been few validation studies performed. Ada95 seemed an excellent vehicle for OO metric validation studies because it is so often used in controlled development situations, but the pure OO metrics are not immediately applicable to Ada95.

In an earlier paper, "Applying Object-Oriented Metrics to Ada 95," (*Ada Letters*, Vol. XVI, No. 5, Sept./Oct. 1996, pp. 48-58), Bill showed how to transform pure OO metrics into measures against Ada95 code involving tagged types and packages. The work also introduced some Ada95-specific OO metrics not found in other OO languages, such as a count of classwide operations.

This presentation focused on results from an initial application of 13 Ada95 OO metrics to data from a small upgrade project involving 2 programmers, 94 classes and about 36 KSLOC.

Results of the study suggest that at least eight metrics have some valid predictive capability. Bill hastened to add that the results are from an uncontrolled study and only provide a single data point. They do, however, suggest that OO metrics for Ada95 could be valid quality predictors and deserve additional study. More controlled studies are planned by DCS Corp.

Yet Another Quote of the Day

Any sufficiently advanced technology is indistinguishable from magic

-Arthur C. Clark, 1962

But this doesn't mean that all magical illusions are underpinned by advanced technology.

-Peter Amey, 2001

Final Thoughts for Ada WOW

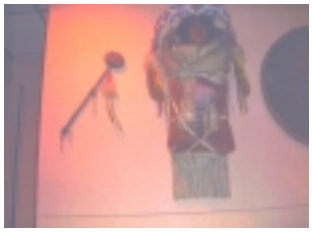
by Currie Colket (Mitre Corp.), SIGAda Chair

I can sum up this conference as WOW, what a conference! It has had some technically significant papers. The report on the UDLP architecture using patterns was very impressive and a real eye-opener for me. This strong start of the conference was carried out with many high-quality papers that will be of value to all of us. Of particular mention was the only paper by a student, Lieutenant Kenneth L. Ehresman, who not only brought a breath of fresh air to our conference, but also is causing Admirals and Navy contractors to take a new look at Ada.

I would like to thank Paul Stachour and the SIGAda 2001 Conference Committee for putting on this conference. Your hard work has really been a great benefit to all of us in the Ada community. We really appreciate your dedication, effort, and capability to respond dynamically to changes forced on us by the events of September 11. I especially thank all of you who were able to fight your organizations' obstacles raised by the terrorist attacks to come to SIGAda 2001. I feel the conference was a tremendous technical success and hope you do as well. I would also like to thank Ann Brandon, the Editor in Chief of Ada WOW, for enabling Hal Hart to walk the halls shouting "Ada WOW, Read All About It!" This record of SIGAda 2001 will be a fine memory for all of us and I would like to thank the many who contributed to making it a success.

I hope each of you were able to find new aspects of Ada that you will be able to use when you get back to your job and establish new contacts that will help you solve the challenges that are facing you. If you did, I hope you tell other folks about the value of the SIGAda conference. I also hope you will want to get involved with the SIGAda community in advancing our technology through our local chapters, our working groups, and our *Ada Letters*. Some of our local chapters and working groups need a jump-start and you could be the catalyst to make them viable again. Our next conference will be in Texas in 2002, pending approval by ACM. If you would like to be part of the organization that puts it on, please let me know at colket@mitre.org. I am looking forward to seeing you there!

Photo Memories of Exhibits, People, Dinner, & the Thunderbird Hotel



Ada's Window on the World Staff

Editor-In-Chief

Ann Brandon (Onyons, Inc.)
(SIGAda Vice Chair for Liaison)

Production Co-Editors
Hal Hart (TRW)
David Harrison(Northrop-Grumman IT)
(SIGAda Vice Chair/Mtgs-Confs)

Contributors:
Currie Colket (The Mitre Corp.)
(SIGAda Chair)
David "Cookie" Cook
(Shim Enterprise Inc.)
Steve Deller (Smooth Sailing LLC).)
Mark Glewwe (Goodrich Corp.)
Robert Leif (Ada_Med)
Clyde Roby (IDA)
(SIGAda Secretary)
Ron Oliver (caress Corp.& TopGraph'X)
Paul Stachour (Stachour Software)
Jeanne Kopesky (webershandwick)

The Comm System Paul Provided Us

SIGAda 2001

Sponsored by ACM SIGAda

THANKS OUR CORPORATE SPONSORS

AdaCore TECHNOLOGIES.INC

Rational the e-development company™

Aonix

Top Graph'X

Ada

emenu Inc. www.emenu-inc.com

DAIMLERCHRYSLER

United Defense