"Ada-WOW" Ada's Window On the World

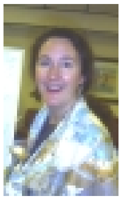
SIGAda 2001
Sponsored by ACM SIGAda

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

Wednesday, October 3
Volume 3, Issue 2

2^{nd} Day Kicks Off with Education

by Ann S. Brandon (Onyons, Inc.)



As the new editor of Ada-WOW, I am truly wowed by the quality and quantity of volunteer writers and walk-by editors that I've been able to lasso into creating today's newsletter. For those who wondered where I was when Currie introduced me as the new V.P. of Liaison at his welcome yesterday morning, my apologies. But I need more than five hours sleep, unlike all the other Ada-WOW staffers.

In today's issue, David Cook comments on the morning's keynote speaker and ends with a single complaint about Scott Edgerton, which he hopes will be corrected at the next SIGAda.

Clyde Roby, Steve Deller, and Ron Oliver cover technical sessions for Ada-WOW and answer all the questions you were afraid to ask with clarity, thoroughness, and occasional wit. David Harrison interviews Program Chair John McCormick about this year's submissions, while Hal Hart writes about the exhibits and the vendor presentations.

Wednesday's sessions kick-off with an emphasis on teaching Ada in a sea of C. Martin Carlisle of the US Air Force Academy will deliver the keynote address, in which he promises to confess his zealotry as a teacher of Ada. Afterwards, speakers explore teaching Ada using everything from robotics (Fagin, Merkle, and Eggers) to compilers (Taft). Tucker Taft also begins the afternoon sessions with a keynote address on how the next generation of Ada can help in "Fixing Software Before It Breaks."

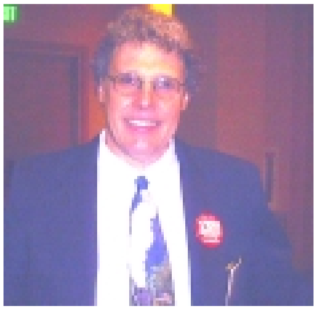
If you want to be in a "Birds of Feather" session tonight from 7 pm-10 pm, then please sign up on the large flip chart in the Hall of Nations.

See you around 9 o'clock.

SIGAda Awards Ceremony at 9 am

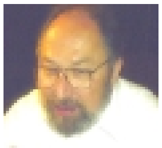
by Hal Hart,
Awards Committee.Co-Chair

Please be sure to arrive on time for the opening plenary session this morning, as the SIGAda Awards ceremony will occur then. SIGAda presents two types of awards – for Outstanding Contributions to the Ada Community, and the ACM SIGAda Distinguished Service Award. Come help us honor this year's award winners.



An Interview with Program Chair John McCormick

by David Harrison



SIGAda's Treasurer, John McCormick *(above)*, is also SIGAda 2001's Program Chair. In a lunchtime chat, he admitted that he had fulfilled his obligations despite the increasing difficulty in encouraging submissions of appropriate topic, broad interest, and highest quality. John stated that he has thoroughly enjoyed working with the members of the Program Committee.

John is a professor of Computer Science at the University of Northern Iowa at Cedar Falls, Iowa. Last spring, he had proposed to members of the management team at Rockwell-Collins in nearby Cedar Rapids that he spend the summer at Rockwell as an Instructor Intern. In effect, practicing what he preached.

John spent eight weeks engaged in consulting and analysis in the Business and Regional Systems groups, on a Canadair Avionics legacy (Ada83) upgrade project. At the end of his self-chosen assignment, he wrote a summary report titled "How I Spent My Summer Vacation." It was so well received, he was asked to present it twice more!

Inside this Issue:

Weather Forecast

Today (Cooler)
High: 21 C (60 F)
Low: 9 C (48 F)

Tomorrow (Cloudy, chance showers)
High: 14 C (57 F)
Low: 4 C (38 F)

Friday (warming to 90)

When John returned to campus life, he found that his experiences had expanded his view of the application of process and principle.

Those who attended the SIGAda '99 conference in Redondo Beach, Calif., might recall John's well-received paper on Ada realtime control of a model railroad. This was developed as part of John's efforts with students in his "Real-Time Embedded Systems Laboratory," which is co-sponsored by Rockwell-Collins and Maytag. (How many lines of code are in the "Neptune" washing machine?). John reported that Rockwell hired one of the two graduates of this course and that more than fifty schools have inquired about John's lab.

Some Ada Trivia

by Ada-WOW

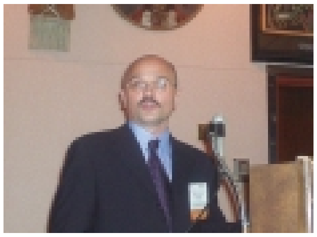
(1) What does the acronym "**ISO**" stand for?

(2) What was the first validated Ada translator?

(3) The first Ada standard was MIL-STD-**1815**. Where did the "**1815**" come from?

(4) What was Ada's middle name?

(See end of issue for answers.)



Roland Trauter Delivers DaimlerChrysler's Vendor Presentation

Greg Gicca (Aonix), above, and his Ravenscar-propelled plane below

A Showcase of Exhibits

by Hal Hart (TRW)

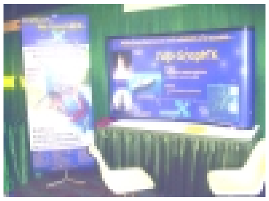
Exhibits opened with a splash at Tuesday morning's refreshment break and continued with strong mid-day vendor presentations by two of the conference's Corporate Sponsors. The first presentation was a pleasant surprise, as we welcomed a newcomer to SIGAda events -- Daimler Chrysler. We have all heard of their automobile sectors but, in the U.S. at least, are less familiar with their European alliance defense/aerospace units. Their Ada cross-reference tool, developed when they found no commercial analyzer met their quality-checking needs, appears to deliver compiler-independent analysis capabilities without using ASIS. Used internally now, they are actively searching a partner to market the tool.

The second presentation was from a long-time Ada community member, Aonix's Greg Gicca. He effectively overviewed the Ravenscar profile and the several Aonix products and many deployed safety-critical systems using Aonix technology. The conference highlight may turn out to be his last chart, an animated plane flapping its wings ala a bird (a raven?).

Meanwhile, six other exhibitors also are staffing booths and demonstrating products in the exhibit hall – including Corporate Sponsors Rational, Ada Core Technologies, and Top Graph'X. Ada Core has a limited presence, including partner DCS, due to severe server and ISP disruptions at their New York City headquarters on Sep. 11; it definitely adds up to a booth worth visiting along with the others. Green Hills Software, Praxis Critical Systems, and TNI are also exhibiting and demonstrating, and two more Corporate Sponsors, United Defense L.P. and Emenu, are showing their support for Ada and SIGAda by renting booths and displaying materials.

Please see the Exhibits Guide (the bright orange four-page brochure) for further descriptions of each exhibitor and their product offerings. Better yet, visit the booths for even more information. We know most of our SIGAda conferees influence those who make purchasing decisions in their organizations. There is no better way to keep abreast of rapidly moving tools and other products than through the first-hand knowledge available in our exhibit hall.

Do not miss the three remaining vendor presentations today. The first starts ten minutes into the morning refreshment break (10:40am) with Ron Oliver representing Top Graph'X speaking on "CORBA for Embedded Systems." Praxis and Rational present their products during the mid-day break starting at 1:15pm.



Top Graph'X Booth

Greg's Animated Chart was So Good We Had to See It Repeatedly

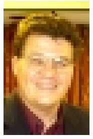
QUOTE OF THE DAY

There are two ways of constructing a software design. One way is to make it so simple that there are obviously no deficiencies. The other way is to make it so complicated that there are no obvious deficiencies.

-Professor C.A.R. Hoare
The 1980 Turing Award Lecture

Ada? WOW!

by David A. Cook
(Shim Enterprises Inc.)



I was indeed privileged on Tuesday morning to attend the keynote address by Scott Edgerton, United Defense, L.P. I had planned on my usual "show up for the keynote address, listen to the S.O.S (Same Old Stuff) about how good Ada is, sleep through most of the talk, and be first in line for the exhibit hall." Boy – was I in for a pleasant wake-up experience.

The title of Scott's talk was "Architecture-based Software Development on the Crusader Program." I had assumed he would talk about the architectural choices and decisions made on the system. While the architectural choices were indeed discussed, Scott gave a talk that basically said, "We are developing an application using state-of-the-art practices and techniques. Ada just happens to be the language of choice for our program."

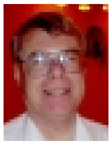
During his talk, Scott covered the following issues: object-oriented design and implementation, child packages, the use of the Unified Modeling Language (UML) and Rationale Rose, automatic code generation from design using an automated tool, realtime timing, interprocess communication, reuse of code based on patterns, abstraction of the hardware from the design, abstracting the timing mechanism away from the programmer, and memory management. Like I said – Ada! WOW!!

In not one case did Scott say that Ada presented a problem. Ada proved to be a natural fit for the project's complex blend of tools, techniques, requirements, and constraints. Scott went on to say that Ada has been a real success for the project. He also pointed out that, contrary to some widely held opinions, the supposed lack of trained Ada programmers was not a problem on the project. Instead, he said that their experience has been that you can teach a good engineer Ada rather quickly by using a blend of textbook training and a mentoring/job training program.

I only had one problem with Scott's talk. Granted, he is an excellent speaker, and his project has obviously found the secret to implementation of a large project using both good practices and a terrific programming language. My problem is this: I am the Tutorial Chair for this conference, and he didn't present a tutorial for us! Based on the enthusiastic response from the audience on his talk, and also based on the fluent way that he answered questions afterwards – I think he could have easily filled a half-day tutorial with his lessons learned and best practices. Maybe next year?

Reengineering Ada95 Using UML

by Clyde Roby
(Institute of Defense Analysis)



Heinz Fassbender's presentation described the experiences of his group in the reengineering of an Ada95-programmed Command and Control Information System (CCIS) by using UML. He updated information since the publication of his paper in the *Proceedings*.

INFIS, the existing research system, is an experimental integration platform for command and control information systems. It is the test bed for the German portion of the Army Tactical CCIS (ATCCIS) study and the Multilateral Interoperability Programme (MIP) with platform-independent access.

Heinz described the INFIS Global Architecture that consists of many domains and several subsystems. Each subsystem has a kernel and a Graphical User Interface (GUI) that uses CORBA to interface to the kernel. There is also a connection between the kernel and a database management system. TCP/IP and CORBA are how the subsystems are interconnected.

UML was chosen to describe the system because it can be understood by many different people, has a nearly formal meaning, is standardized by the Object Management Group, and CASE tools can automatically generate code from the UML database.

Heinz then described the reengineering process with UML diagrams using Software-thru-Pictures (StP). In this process, the model was separated into a high-level and a low-level model and then was recombined into a single static model. From this static model and a dynamic model, requirements were extracted for the development of an updated model. Finally, this was used to restructure the INFIS model and to redesign the system. New requirements could then easily be added for this system.

The INFIS high-level global architecture (UML model) contains abstract classes with appropriate relations and associations. He displayed the more expressive UML model for the kernel, the low-level inheritance tree in a macroscopic view, and the low-level detailed UML model. Both of these low-level models are automatically produced by the reengineering component of the CASE tool StP/UML. This UML diagram effectively completes the static structure of INFIS.

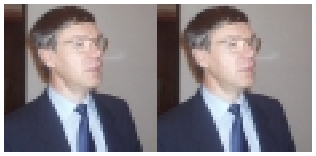
Sequence diagrams are used to define the dynamic model for INFIS (this figure is very similar to the one shown in the Proceedings).

The requirements are extracted from the models above. The tool's use-case diagrams then further model the requirements.

At this point in the process, Heinz explained, there are two possible alternatives to further development: (1) redesign the complete system such that only the application level has to be extended, or (2) use new techniques and buy an application server. The first alternative implied the necessity of building an application server for the project. The team chose the second alternative: use new techniques and purchase Java 2 Enterprise Edition. This does not mean that they would have to change the reengineering process. However, instead of reimplementing the system in Ada, they would do it in Java.

In conclusion, Heinz said that they generated good documentation of the system, extracted the necessary requirements, and decided to reimplement the system anew in Java. They will replace CORBA interfaces by either Remote Method Invocation (RMI) or by Java DataBase Connectivity (JDBC); the new GUI will also be Java based.

Heinz's presentation was very understandable, especially using well-developed UML diagrams of their reengineering process for INFIS.



The 2 Speakers in Tuesday's 2nd Session

Developing High-integrity Software

by Steve Deller (Smooth Sailing LLC)

Peter Amey *(see photo above)* of Praxis Critical Systems did a yeoman's job of giving back-to-back presentations on Tuesday. He was filling in for Ben Brosgol of ACT in New York City who was unable to make the conference trip because of the Sept. 11th attacks on the city.

Peter's first presentation was on how SPARK proof annotations have evolved from being "about code" to being "about abstraction." SPARK annotations are used to strengthen Ada specifications sufficiently to allow static analysis and proofs to proceed even before implementation. The addition of abstract annotation variables and modes for annotation variables has created a much more powerful analysis system.

Peter showed how SPARK-annotated Ada specifications can now be used for proof analysis of entire system properties instead of just software properties.

Peter's second presentation, "Logic vs. Magic in Critical Systems," focused on software process mindsets that can benefit or hinder good software engineering. This talk was an overview of his paper of the same title published in the *Ada Europe 2001 Proceedings*. A copy of the full paper is on the www.sparkada.com website.

An example of a software process mindset is Edsger Dijkstra's "Goto Considered Harmful," which benefited software engineering by being the catalyst in the use of structured programming. But it also hindered software development when companies tried the "magic" of "de-spag" tools, which eliminated goto's automatically and were therefore *also* to cough up "de-spaghettied" code. The structured programs that result from this "magic" are less readable and maintainable than the original spaghetti.

We have seen a more recent example of "magic" hindrance in the deification of "object-oriented" programming methods. No one doubts the benefit of object reasoning in the application domains. The hindrance comes from pushing this as the "magic" solution for software, particularly safety-critical software.

To reach the very high-assurance levels required by safety-critical code, we must effect a qualitative change in how we build software. It is insufficient just to "do it more carefully." To reach correctness by construction, we must show programs are safe before testing through logical reasoning about program information. Peter points out that what we often call *information hiding* is really *detail hiding* through abstraction.

In particular, state information is the single biggest factor in analyzing a program's cohesion and coupling. Yet OOP, the "magic" extension of reasoning about objects, considers state information an implementation detail. UML provides no notation at all to express state. The result is that object orientation, when taken to excess as "magic," actually hinders the development of safe (reliable) systems. The end result of OOP taken to excess is spaghetti code in the large, or what Peter calls "macaroni code."

Peter's background in aviation showed through when he quoted a statement from the engineer responsible for the extremely successful Ford Trimotor ("the Tin Goose"): "Simplicate and add lightness." An excellent prescription for software.

Peter provided ample evidence that logical reasoning about software leads to much higher quality at much *less* cost. The Lockheed C130J software upgrade program review stated that using SPARK and formal analysis methods resulted in "Less than one-fifth the normal cost of the industry." When subjected to rigorous hand-analysis of code: "SPARK code had ten percent of the residual errors with full Ada, and full Ada code had ten percent of the residual errors in C code."

And finally, Peter revealed why he has been driven to specialize in Ada rather than in any other high-level language:

"A superior pilot uses his superior judgment to avoid those situations that would otherwise require his superior skill."

Another Quote of the Day

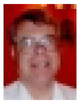
Real-life problems are those that remain after you have systematically failed to apply all the known solutions.

-Edsger Dijkstra, 1973

Rational's Booth

Dead Reckoning Tracer Aids the Deck Officer

by Clyde Roby



This program was the result of a Masters Thesis at the Naval Postgraduate School by Ken

Ehresman and his colleague, Joey Frantzen. Today, Navy admirals are interested in putting their program on all ships in the fleet. Its graphical display output replaces the manual hand-drawing of graphs that had been developed on paper using grease pencils and rulers. Ken and Joey saw that there was a need to automate this paper-driven process so that accidents would decrease on the sea, where ships come very close or actually bump into one another.

Their program had to be a stand-alone reliable system that could easily be placed on every ship and not just on warships. It had to use COTS products and it had to tie into the LAN systems on every ship. The primary goal, when using their program, is to minimize the number of collisions in the Navy by overcoming the main cause of such collisions: distracted Navvies.

The program was originally developed in C++ but was rewritten in Ada so that it is now hardware- and operating-system independent. It does not use Java. Ken and Joey tested the code under both the Linux and Windows 2000 operating systems; it had to run exactly the same on both.

The overall software design uses UML's rigorous, traceable, and maintainable models. The development uses a certified Ada compiler and environment, thus making the code robust. The "code does what it's advertised to do."

The UML model view controller allows for flexibility and multiple views of the same data. It is highly extensible and robust, meets emerging needs, and runs on multiple platforms.

The program was developed using GtkAda, GNAT, and GVD (a GUI-based debugger), as well as the Gtk+ toolkit. This Ada development environment supports pragma C and pragma FORTRAN, thus allowing the program to interface with previously written code. It also supports OpenGL.

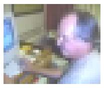
Benefits derived from using this program will include reduced manning onboard ships, more accurate calculations, faster information display, faster distribution of information, portability, and robustness (using distributed databases, which are also written in Ada). This all equates to fewer collisions, thus saving many lives and dollars.

Future enhancements could include wireless LAN connectivity; the use of touch screen displays, voice recognition technology, mobile headset/communications; automated deck log entries, Palm Pilot/CE devices for information on demand; and integration of multiple views (Falcon view, CA-DRT, heads-up displays, etc.).

As mentioned earlier, the prototype system was developed in C++ but was rewritten in Ada because there was a lot of "chasing compiler implementation problems of C++ constructs." The GUI was dependent upon Microsoft's MSC, and the prototype was not portable. The code output from the Ada compiler "ran as advertised."

After describing the program and how and why they decided to use Ada for the major redevelopment, Ken then gave a demonstration of their Ada-coded prototype.

Ada95 Bindings for the NCSA Hierarchical Data Format



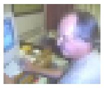
by S. Ron Oliver (Top Graph'X & caress Corp.)

Bruce Barkston (NASA at Langley) reported developing Ada95 bindings for HDF4 and HDF5, which are the current versions of the National Computational Sciences Alliance (NCSA) Hierarchical Data Format (HDF). These self-describing file formats are intended for storage of large, diverse collections of scientific data and for retrieving subsets of these data. The technology has applications across a wide range of scientific and engineering domains.

Bruce is doing this work in conjunction with a project for which he is the Principal Investigator: on-board satellite systems that gather information on solar radiation and related information that may be useful for better understanding weather patterns. HDF is of particular interest because its self-documenting feature will be invaluable for a data-collection exercise that may well span more than a century.

The final results of this work will be reported in a public domain NASA document, and we hope will be made available on the Web.

Automating Software Module Testing for FAA Certification



by S. Ron Oliver

Usha Shanthanam told us of Boeing's (Wichita) experience in developing tools and techniques to facilitate their efforts to achieve FAA Certification for software with respect to DO-178B. DO-178B requirements entail, among other things, full-coverage testing. Doing so manually can be onerous— tedious, time-consuming, cumbersome —and hence error-prone.

At Boeing they decided to develop tools to automate the testing process. Test Set Editor (TSE) consists of a proprietary set of Ada programs, which work in combination with a spreadsheet program and homegrown test scripts written in Tcl/tk. An engineer, who need not be a programmer, defines test cases in a spreadsheet. The Tcl scripts use the spreadsheet data and TSE programs to generate Test Drivers.

The Test Drivers execute without need of operator input or intervention, and produce a detailed report, including an assessment of the percent of full coverage achieved. With the spreadsheet data, Tcl script, and automatically generated Test Drivers, the tests are fully and precisely repeatable.

Data collected indicates that using TSE reduces the scope of effort to do the testing can be reduced by a factor of 5 to 10. The more complex the code, the higher the reduction rate.

A Scene from Tuesday's Conference Dinner

Lessons Learned from the SIGAda 2001 Chair

by Paul Stachour,
SIGAda 2001 Conference Chair

I'm a veteran Ada community member. I started by reading "Ironman" in SIGPLAN notices, where I liked what the language requirements were. I had already seen that having poor materials (highly deficient languages) was a major cause of errors. Also, I was so much more productive when the language helped me.

I was involved with the Green Team: a small portion of the design for the calendar/duration/time package is mine. I was the tutorial chair (6 ½ day sessions with roughly 75 people each) for the 1985 SIGAda conference here in the Twin Cities.

Other than that, I haven't been involved in organizing Ada events. So I was surprised when Currie called me in 1999 and suggested that the Twin Cities would be a good place for a SIGAda conference, and that I would make a good conference chair. I said, "You're kidding!" He talked me into writing up and submitting a proposal.

I started slowly in 1999, when I got on some of the email lists as a "lurker."

We did a proposal for 2000, but Baltimore was chosen instead. At the 2000 Baltimore conference, we got an OK to hold it here in the Twin Cities. I felt a lot more confident when several of the EEC members said mine was the most complete proposal they'd seen in a long time.

In putting together this conference, I can say that the biggest surprise was the conference budget. There were a number of constraints that I didn't know about. I kept refiguring and rewriting and resubmitting the budget before we got approval to "Officially Start." In addition, since it was 10 ½ months instead of 12 between conferences, time was short.

My biggest disappointment was not producing a musical for Tuesday night. When we were told that a sponsored musical was not in the works this year, we had to find something else to fill the void.

We tried to find something that would appeal outside of the Ada community, and that we could do jointly with other societies. Unfortunately, it took longer than we had hoped to find a topic and speaker. We finally decided on a combination dinner/presentation.

This left us much less time than we had hoped to do the joint marketing effort.

I wish we had had longer to market; I think we could have gotten a much larger number of people from outside of the Ada community.

My biggest unilateral decision was in choosing the Thunderbird Hotel. Besides the décor (we wanted to get away from the usual hotel blasé), we were able to negotiate a contract with a good upside and a small downside. My gamble paid off. In light of Sept. 11's terrorism attack, which resulted in a reduced number of late reservations, and my breaking my leg a month before the conference, which gave me less energy for marketing and organizing, this turned out to be a good choice.

Paul's Deputy Conference Chair,
Jan McArthur

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)

Contributors:
Currie Colket (The Mitre Corp.)
David "Cookie" Cook (Shim Enterprise Inc.)
Steve Deller (Emenu Inc.)
Mark Glewwe (Goodrich Corp.)
David Harrison(Northrop-Grumman IT)
Robert Leif (Ada_Med)
Clyde Roby (IDA)
Ron Oliver (caress Corp.& TopGraph'X)
Paul Stachour (Stachour Software)
Jeanne Kopesky (webershandwick)

Trivia Answers

(1) *Trick Question* -- "ISO" is *NOT* an acronym. "ISO" is a trademark for the International Organization for Standardization, and supposedly not an acronym in any language.

(2) AdaEd, developed at NYU by Ada Core Technologies founders Robert Dewar, Ed Schonberg and others, was the first validated Ada translator. (*April 1983)*

(3) 1815 was the year of Ada's birth.

(4) *Trick Question* – Ada's middle name is *"Ada."* Her first name was Augusta; Ada was her middle name.