

AEROSPACE

engineering & manufacturing

Technology Update

October 2008

Avionics

Conflict-detection technology takes to the skies

The U.S. FAA instituted the Free Flight program to better enable controllers to handle significant growths in air traffic, and to increase safety and capacity at Air Route Traffic Control Centers (ARTCCs). This program included the development and deployment of the User Request Evaluation Tool (URET), conflict-detection technology that automatically detects and advises air-traffic controllers of predicted conflicts between aircraft and special-use airspace within the National Airspace Systems.

URET is based on a prototype that was developed by **Mitre's** Center for Advanced Aviation Systems Development. It can determine whether pilot-requested changes in flight plans are free of conflicts with other air traffic. It can also evaluate pilots' requests to deviate from their planned routes to avoid adverse weather, assign more direct routing of aircraft, and evaluate changes in altitude to take advantage of favorable winds. Prior to URET, controllers relied on paper flight strips and mental calculations to determine whether a proposed route change would be conflict-free.

The FAA awarded **Lockheed Martin** the multi-million dollar contract to fully develop URET, including systems engineering, software development, integration and deployment, hardware design, and installation.



The User Request Evaluation Tool developed by a team that included Lockheed Martin and AdaCore is conflict-detection technology designed to save time, fuel, and money while also helping to ensure safe aircraft separation.

After an extensive phase study, Lockheed Martin selected **Sun Microsystems'** SPARC processor architecture as the computer platform for URET, and Ada was selected as the preferred language for software development. To support software development for the platform, the development team needed an Ada compiler.

Lockheed Martin had very specific requirements that warranted more than standard compiler deliverables. First, the development team wanted to reuse some existing software products for URET that were originally developed on different computer platforms. Second, it wanted to use the POSIX thread library of Solaris for protection from priority inversion to

ensure the correct behavior of key, real-time algorithms used in several of Lockheed Martin's air-traffic control applications. In addition, Lockheed Martin required a compiler that could deliver correct signal and interrupt handling for URET's multiple-threaded and multitask environment.

Following an extensive, robust evaluation period that included comparing function, cost, and support, the Lockheed Martin URET team selected the GNAT Pro development environment and ASIS-for-GNAT products from **AdaCore**.

Because Lockheed Martin wanted to reuse some existing software products developed on different computer platforms, a type dictionary support tool was

needed to port from one platform to another. The type dictionary lays out the details of all fields of a particular Ada type at the bit level. It was implemented using AdaCore's ASIS-for-GNAT, which is an implementation of the Ada Semantic Interface Specification (ASIS).

ASIS is a library that gives applications access to the complete syntactic and semantic structure of an Ada compilation unit. This library is typically used by tools that need to perform some sort of static analysis on an Ada program.

ASIS is an international standard (ISO/IEC 15291:1995), and is designed to be compiler-independent. Thus, a tool that processes the ASIS representation of a program will work regardless of what ASIS implementation has been used. This means, for instance, that most ASIS applications can run on a different target than the final target.

However, for some kinds of utilities, it is useful for ASIS applications to be able to deal with target-dependent issues, such as the size of floating-point types or the layout of record types. The Data Decomposition Annex (DDA) of ASIS is designed to provide just that kind of specialized target-dependent information, which can be useful when using ASIS in conjunction with embedded applications. ASIS-for-GNAT fully implements the DDA, unlike other competing ASIS implementations, which omit this capability, according to AdaCore.

To support the type dictionary, the Lockheed Martin URET team also needed the optional ASIS DDA to provide the exact layout of the bits to port this tool from Lockheed Martin's previous platform to SPARC Solaris and the GNAT Pro underlying data structures. As an optional annex, the team requested that AdaCore add this capability to its ASIS-for-GNAT implementation.

Lockheed Martin uses the type dictionary and the ASIS interface for more than just interrogating the compiler about deci-

sions it has made about data layout on various platforms. The technology also is used to take data recorded online, which would normally just look like a stream of bits, and process that data offline. The dictionary content literally lays out a map of the individual type and allows interpretation of those bits in a context-specific and meaningful fashion.

The type dictionary also enables platform independence, one of URET's key requirements. For example, in another system Lockheed Martin has two subsystems, one executing on a Solaris operating system with a GNAT Pro compiler, and the other subsystem executing with a different compiler and platform. The URET team needed to use representation clauses to impose the exact layout to follow on each platform. It used the type dictionary to come up with a common definition that could be used on both subsystems.

In addition to the ASIS DDA, AdaCore was also asked to enhance the GNAT Pro compiler to support the POSIX thread library of Solaris. Lockheed Martin's requirements were stringent regarding protection from priority inversion, and its URET team requested that certain thread priorities be available to make proper use of the POSIX thread library.

Priority inversion is a problem in concurrent systems with shared resources. Typical priority inversion occurs when a

high-priority task waits for a resource that is currently held by a low-priority task, but the low-priority task has been interrupted by a medium-priority task that is unrelated to the first two tasks. Thus the high-priority task may be delayed for an arbitrary amount of time waiting for the low-priority task to restart.

Lockheed Martin also needed changes in the way signal interrupts were handled. In certain multiple-threaded or multitask environments, the wrong signal was being delivered to an unexpected process or task in the executable. This proved to be a particularly complex problem to solve, as it was intermittent.

During the development and testing stage, AdaCore solved the problem when it realized that the URET team needed support for three types of handlers: the system, runtime, and user. To support these needs, AdaCore introduced a new "pragma Interrupt_State," which allowed program developers to clearly map interrupts or signals to a default system handler, the Ada Runtime and an Ada exception, or to a user-defined handler using an Ada-protected procedure or interrupt entry, respectively.

The FAA ended up being able to install URET systems at all 20 ARTCCs on or ahead of schedule. Since URET was deployed, the FAA estimates that carriers have reduced routes over 89.5 million nmi.

Jean L. Broge



North American Headquarters

104 Fifth Avenue, 15th Floor
New York, NY 10011
tel +1 212 620 7300
fax +1 212 807 0162

European Headquarters

46 rue d'Amsterdam
75009 Paris, France
tel +33 1 49 70 67 16
fax +33 1 49 70 05 52