

# Computers in engineering

## Finding bugs and saving time

A flight management system (FMS) is used to guide commercial aircraft to their destinations, so a high level of reliability and availability is vital. In previous development efforts, **Smiths Aerospace** used static and laboratory testing to identify runtime errors. However, the software powering the new FMS for the **Airbus A319/320/321** and **A330/340** families was so large that an enormous effort would have been required using traditional testing methods.

With potentially a number of difficult runtime errors still to be isolated during system integration, Smiths recognized that improved testing methods would have to be adopted. During technical interchange meetings with Smiths, engineers from Airbus founding partner **EADS** recommended abstract interpretation. This technique automatically checks the dynamic properties of software applications without executing the program.

"Identifying these errors with lab testing would have been very difficult because we would have had to recreate the errors and then capture the state of the machine at that moment," said **Dudrey Smith**, Manager of Support Software for Smiths Aerospace.

**Thales/Smiths'** new FMS for Airbus, called **FMS2**, provides the Airbus **A320** and **A340** families with a true second-generation flight management system, including every feature needed to meet the latest air-traffic regulations, while improving the aircraft's operability and performance. The **FMS2** will meet airlines' forward fit as well as retrofit needs, and provide the right tactical tool for Communications/Navigation/Surveillance with Air Traffic Management (CNS/ATM) capability.

"We had most of the code in place and were going through extensive traditional high-level software testing and system-level lab testing," said Smith. "The project was challenging because, with half a million lines of code, this

was one of the largest programs we had ever developed. We created a flight simulation environment in the laboratory and began identifying runtime errors. In a typical case, a line of code would run perfectly 99% of the time and then fail all of a sudden. We estimated that we had hundreds and hundreds of hours of troubleshooting left to be able to assure our customers that our code would never let them down."

Smiths originally initiated an internal "improved integration testing methods" project through its continuous improvement program. One of the aspects of this activity was the evaluation and integration of abstract interpretation tools for the purpose of identifying runtime errors before lab integration. As part of the evaluation activity, **EADS** was contacted and provided a strong recommendation for the tool.

**EADS** was a lead contractor in the **Ariane 5** European rocket launcher that in 1996 suffered a major failure in its inertial guidance system. An independent board of inquiry took urgent steps to better analyze the software embedded in the inertial guidance system so that such failures would be avoided in future launches.

**Aerospatiale** enlisted a young researcher at the French **INRIA** (Institut National pour la Recherche en Informatique et en Automatique) who devised a software program to identify runtime errors. The software was instrumental in completing the software analysis. The program later was commercialized under the name **PolySpace**.

**PolySpace** represents a dramatic departure from current software testing tools, which are only capable of detecting a certain fraction of runtime errors. For example, test case generators evaluate possible scenarios defined by the quality control staff but cannot evaluate the reliability of the application under every possible input condition. **PolySpace** overcomes these limitations by using abstract interpreta-



**Smiths Aerospace used PolySpace software to analyze its flight management system's software.**

tion techniques to check the dynamic properties of a software application without actually running it and without requiring any test cases. Instead of iteratively analyzing software states like a case generator, it works on an abstraction of the source code. This technique eliminates the need to check every possible value of every variable, which makes it practical to analyze source files as well as programs and to pinpoint runtime errors as soon as the code becomes available.

The **FMS2** code would have provided a challenge for any automated verification method because of its size and the fact that it includes many "pragmas"—user-defined commands that extend the Ada language. With a bit of assistance from the **PolySpace** technical support staff, Smith and his team were able to get **PolySpace** for Ada to analyze their full application on a 1.3-GHz personal computer with 1 GB of memory. After some adjustments and tuning, the time to analyze the entire 350,000 lines of code was about 30 hours.

"The first time we ran the software it pinpointed all of the remaining runtime errors that our testing staff had been

working on," Smith said. "These included three serious concurrent access errors in which the database was being updated at the same time that data was being used by another process. Besides identifying these errors, PolySpace for Ada also provided an excellent dynamic calling tree and

global data dictionary that were very useful in fixing them. These errors would have taken a long time to isolate with laboratory testing."

The latest version of FMS2 was recently certified to DO-178B, an international development and testing standard and is currently in service on the

Airbus single-aisle fleet. According to Smith, the final phase of the project was particularly productive because time that had been previously devoted to identifying runtime errors could now be dedicated to other tasks.

David Alexander

## The PLM advantage

By trimming costs, enhancing product reliability, and maximizing component commonality between product designs, avionics and defense contractor **Kollsman** is winning business. Its ability to deliver cost-effective, reliable, and state-of-the-art defense and avionic solutions comes from its commitment to continually refine product data and processes to develop new and more innovative product with increasing speed and accuracy.

its ability to manage the complex Pro/E model hierarchy. Comprehensive, out-of-the-box capabilities were important to us. Instead of initiating a long, costly consulting agreement, we were up and running with ProductCenter—vaulting, revision/release control, and the ProductCenter Pro/ENGINEER Integrator—in just three weeks."

ProductCenter quickly allowed Kollsman to centralize, track, link, and more effectively reuse not just CAD

creation of viewable **Adobe** PDF files of product data. This allows data to be accessed or printed from any computer platform, not just from the authoring application. Now, over 120 people throughout Kollsman have secure, instant access to data through ProductCenter.

Another cost and time saver has been Kollsman's use of ProductCenter Workflow to handle engineering change orders (ECOs). Collating and copying supporting materials, hand delivering ECOs, and following up to make sure documents kept moving could take weeks or more. Now, an engineer simply fills out an electronic ECO online in ProductCenter and attaches supporting data, and the ECO is automatically routed to a specific person, role, or group of people, often in parallel to save time. Through ProductCenter Workflow, the ECO is routed via e-mail and appears in the reviewer's in-box for them to claim. Reviewers can comment on the form, return it for additional detail or refinement, or approve and pass it along. The ECO's progress is tracked in ProductCenter, with bottlenecks immediately identified. The process now takes just days.

Next on the agenda for Kollsman is the integration of ProductCenter bill-of-material (BOM) to the company's Oracle Manufacturing ERP system. The goal: to use ProductCenter to automatically populate a BOM from Pro/ENGINEER and transfer BOM master data directly to the ERP system. This will not only save time, eliminating keying and re-keying of BOM data between systems, but also improve data accuracy between the systems, streamlining release to manufacturing, and preventing costly errors downstream.

David Alexander



**Kollsman's implementation of ProductCenter PLM software from SoftTech has delivered timing advantages on a number of programs, according to Gregg Wilder, Manager of Design Engineering.**

Like many companies, in the mid-1990s Kollsman managed engineering design data manually with paper and microfiche. The company was using a homegrown data management system, but the solution became unworkable with an upgrade to an **Oracle** manufacturing enterprise resource planning (ERP) system and Y2K system updates.

"We needed a new product life-cycle management [PLM] solution that would plug-and-play with our design tool of choice, Pro/ENGINEER [from **PTC**]," said Gregg Wilder, Kollsman Manager of Design Engineering. "We chose ProductCenter [from **SoftTech**] for

files, but all related product data, including proposals, test procedures, manufacturing method sheets, and ISO procedures.

"Since recently being deployed enterprise-wide at Kollsman, ProductCenter has delivered timing advantages on a number of programs, including our state-of-the-art Enhanced Vision Systems, Digital Air Data Test Sets, and Digital Altimetry systems, all of which use Pro/ENGINEER and the ProductCenter Pro/ENGINEER Integrator," said Wilder. Kollsman has also implemented an easy-to-use find/view/print application using ProductCenter GenView to facilitate