

# Testing Times

April 2000

The IPL Software Products Newsletter

No 15

## Unit Testing Really Works!

**It's official - unit testing really works. A team from the Carnegie-Mellon Software Engineering Institute (SEI) has recently shown that Unit Testing was very effective in reducing error densities in the F-22 software programme.**

The study concentrated on the 1.9 million lines of Ada code being produced by Lockheed Martin and other contractors as part of a 'best practices' program. The results showed that the sampled blocks of code contained, "far fewer errors per 1000 lines of code (KLoC) than the industry standard". According to SEI metrics, the industry standard would be regarded as 5.5 errors per KLoC, while the most recent block of F-22 software was averaging less than 2 errors per KLoC.

So, to what does Lockheed Martin attribute this achievement? One of the company's engineers commented, "We can attribute low error rates to our process and the use of AdaTEST (for unit testing)."

While not wanting to claim that AdaTEST is the answer to "life, the universe and everything", we think that the message is clear: achieving this level of software reliability is only possible by the use of unit testing to produce high-quality software components.



*The F22 'Raptor'*

In the same vein, the EuroSTAR'99 conference last November provided some interesting conclusions. Here's one we like, "After two significant problems in a single module were

detected (with Cantata), the development staff became ardent advocates of tool-based unit testing, firmly believing that the heavy investment... was fully justified." IAI Israel.

### IPL Supports Beagle 2

**IPL is pleased to announce that it will be supporting the UK space effort by supplying copies of AdaTEST to the Beagle 2 software development teams.**

For the benefit of those who do not regularly watch Blue Peter, Beagle 2 is a lander intended to explore the surface of Mars. It will be launched on the ESA Mars Express mission in 2003. The purpose of Beagle 2 will be to answer the fundamental question of whether



*The Beagle 2 Lander*

there was, is, or could have been, life on Mars? See the project web site [www.beagle2.com](http://www.beagle2.com) for further details.

The role of AdaTEST in this prestigious venture will be to assist the software developers in producing the necessary software to allow Beagle 2 to land and operate reliably. IPL is also supplying Beagle 2 with a metric/imperial conversion table, just in case of problems in this area...

On this very subject, the Mars Climate Observer (MCO) mishap investigation

has published its latest report on the loss of the mission. Amongst other things, it blamed, "inadequate independent verification and validation of the MCO ground software." (Source: Aviation Week, March 20 2000). For those people thinking that only airborne software needs testing, think again!

### Metrics for Object-Oriented Software

*In this short article, Misha Dorman - the architect of IPL's software testing products, explores the benefits of OO software metrics.*

Software code metrics have been much touted as an aid to estimating the quality and reliability of code. Even for procedural languages like C, there has not been a great deal of agreement over which metrics are useful. For OO languages such as C++, the situation is even less clear. This article attempts to shed some light on the matter and gives guidance on selecting OO metrics.

Metrics used for traditional (procedural) designs range from simple counts (such as the number of lines of non-comment source code) to more elaborate measurements of the control flow between software elements (such as cyclomatic complexity). Several studies have been conducted, which confirm the intuitive assumption that larger, more complex software is likely to have more faults and will take longer to test. The data obtained by these studies have been used to calculate industry-average coefficients, which have allowed project managers to use metrics to estimate defects and test effort. They can produce even better estimates by performing their own local studies and using the data to calculate project and team-specific coefficients.

*(continued on page 3)*

**Inside: Data Systems & Solutions  
- Cantata++ Case Study**



# Cantata++ - Many Uses!

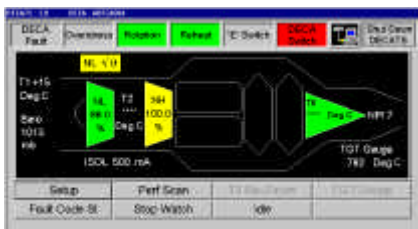
## Case Study from Data Systems & Solutions

For the case study in this issue of **Testing Times**, we profile some recent work from a world-leading independent software safety and verification specialist, the **Software Engineering and Assessment Services (SEAS)** group of **Data Systems & Solutions**.

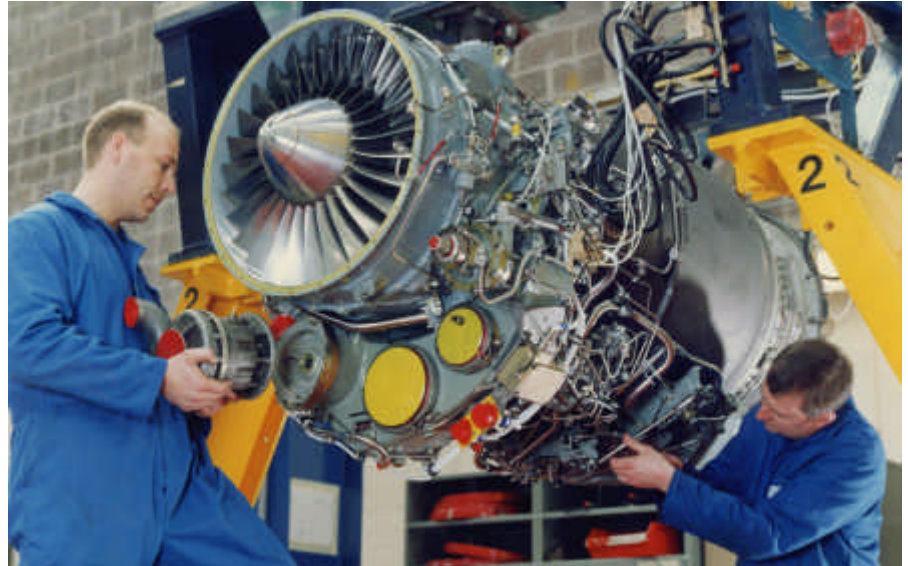
Data Systems & Solutions was formed in early 1999 as a joint venture between Rolls-Royce plc and Science Applications International Corporation (SAIC). It brings together complementary businesses which draw on the project management, engineering, information technology and systems integration skills of its two parents. This combination of experience and resource has produced an exceptional new company, focused on delivering high-quality, world-beating information technology products and services. The SEAS business unit within Data Systems & Solutions is at the forefront of high-integrity software systems engineering applications. SEAS' work with AdaTEST has been previously featured in a *Testing Times* case study (see <http://www.iplbath.com/p38.htm>). In this issue we look at its use of Cantata++ on three C++ projects.

### Rolls-Royce Adour - DECATS

The first project was to supply a test system for the Rolls-Royce Adour Mk 106 gas turbine engine which powers the Jaguar strike aircraft. The project was known as DECATS (Digital Engine Control Amplifier Test Set). It takes the form of a ruggedized PC which can be linked via an RS-422 port to the engine controller. The DECATS extracts data mainly in the form of engine temperatures and speeds, and then performs a variety of sophisticated analyses and tests. Examples of these include calibration of cockpit instruments and timed engine accelerations. The DECATS can also be used to perform adjustments to key



DECATS Screen Shot



*The Adour Engine - Used in the Jaguar Aircraft*  
engine datums and for the off-line analysis of engine fault code data.

The software for DECATS was produced entirely in C++ by Data Systems & Solutions, and presented certain challenges in its verification. The system was assigned a Safety Integrity Level of 'C' according to the DO-178B standard. This equates to a 'Major Failure Condition' if the system goes wrong. The standard requires that the software should be unit tested and evidence of coverage at statement and branch levels provided. In fact, in the interests of efficiency, the DECATS software architecture was partitioned into 'Safety-Related', 'Other Application Modules', and 'MFC' classes (mainly relating to the user interface). A decision was made to fully unit-test all safety-related modules, and to selectively unit-test the other code modules depending on their complexity. This was judged by reference to recognised complexity metrics, and how algorithmically detailed the modules were.

Mike Ferguson was the Data Systems & Solutions engineer tasked with selecting a suitable tool for the verification activities. According to him, "Previous good experience with the IPL AdaTEST and Cantata tools to evaluate". Evaluation and training were carried out during late 1998, and Cantata++ was selected for the project in March 1999.

Two specific features of Cantata++ proved very popular with the engineers. Simon Prior particularly praised its ability to get inside the classes being tested using the 'Friends' technique so that internal ("whitebox") data could be both set and checked. The other key Cantata++ facility was 'Wrapping', which allowed a class to be tested independently of all other class interfaces. Prior says, "Without this, difficult external conditions not directly influenced by our code, such as memory allocation errors, may have been difficult to simulate."

In terms of testing strategy, the Data Systems & Solutions engineers are very happy to endorse the general principle of testing C++ classes by hierarchical integration. They also support the Harrold principle\* which requires re-testing of base functionality when testing in the context of a derived class. In addition to unit testing, Cantata++ also proved useful during system testing, by measuring code coverage. This was carried out on the entire system and helped identify 'gaps' in the system test sets. A surprise result of the coverage analysis was to expose some lines of redundant code that were later removed.

Prior also praised both the usability of the tool and the service provided by IPL. "The integration of Cantata++ within the Microsoft Visual C++ environment was pleasing and easy to use, and the technical support team

were 100% effective in terms of speed, accuracy and general helpfulness.”

**Railway Signalling - Westinghouse**

The ability of Cantata++ to provide coverage and metric analysis also came in useful in the second example. This was a project to conduct system acceptance tests on a railway signalling configuration tool for Westinghouse Signals. The customer had provided the test specifications, but Data Systems & Solutions were required to provide code metrics, determine the coverage and report any areas of code not exercised. This was previously undertaken using a number of proprietary tools, but Cantata++ was able to provide all of the necessary features in one integrated package that proved easier and more cost-effective to use.

Once again, the system was in C++ running on Windows. The product being tested had a complex GUI, so the testing itself was driven by a GUI Test tool, with Cantata++ running in the background. Jason Pinn, the engineer involved with the Unit Testing, reports “Although some minor problems were encountered with Cantata++, the IPL support staff have been very good at providing timely solutions”.

**Nuclear Reactor Test System**

For the final Data Systems & Solutions project, we turn to the nuclear industry. The company was awarded a contract to carry out an independent assessment of the safety systems for a nuclear power plant. Part of this work has involved the construction of a Dynamic Test System (DTS) for a Primary Reactor Protection System (PRPS). This work has been carried out in C++ using a development process guided by the IEC 880 software safety standard. Once again, Cantata++ played a crucial role at various levels: the static analysis metrics helped Data Systems & Solutions engineers identify the modules requiring special attention, and coverage analysis of the system tests allowed them to complete the work to a high level of confidence.

The principal Data Systems & Solutions engineer in charge of development work was Karl Aspinall. He comments, “The use of Cantata++ has proved very worthwhile. We have experienced no problems with it, and that is what you need from a test tool.”

*IPL is grateful to the Data Systems & Solutions SEAS team for their participation in this Case Study. We wish them well with their ongoing work.*

*\*For more information on the Harrold concepts on testing OO/C++ software, see the IPL paper at <http://www.iplbath.com/p833.pdf>.*

**Metrics for Object-Oriented Software**

*(continued from page 1)*

However, both academic studies and practical attempts to apply such traditional metrics to object-oriented software have been less successful. Although the strong correlation between software size (measured in lines of source code) and number of faults is still observed, albeit with different coefficients, the correlation between the standard complexity metrics and defect density is far less pronounced.

The problem is that the complexity of object-oriented software designs lies not just in the control flow between statements within a routine, but also in the flow of control between routines as a result of polymorphic method calls. Traditional complexity metrics simply do not attempt to measure these aspects of the design. In their place there have sprung some additional metrics specifically designed to measure the complexity of object-oriented software. These include the MOOSE (Metrics for Object-Oriented Software Engineering), MOOD (Metrics for Object-Oriented Design) and QMOOD (Quality Metrics for Object-Oriented Design) sets.

As with traditional software metrics, different metrics measure different features of the software design. In order to make best use of object-oriented software metrics it is important to understand these differences. In the explanation that follows the metrics' names in brackets are those used by Cantata++.

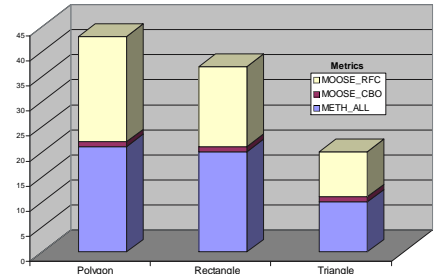
Simple metrics such as methods defined (MOOD\_MD), attributes defined (MOOD\_AD) and weighted methods in class (MOOSE\_WMC\_MCCABE) can be used to measure size and thus provide an initial estimate of the expected number of faults. These metrics are less susceptible to the vagaries of local (or personal) code layout styles than the traditional number of lines of non-comment source code.

Metrics such as hierarchy quality (MCCABE\_QUAL), lack of cohesion among methods (MOOSE\_LCOM98) and attribute hiding factor (MOOD\_AHF95) can be used to highlight poor or questionable designs. A non-zero value for hierarchy quality indicates the presence of a dependence from a base class onto one or more of its

derived classes indicating a problem in the inheritance hierarchy. A high value for LCOM may indicate a class which attempts to represent more than one concept which should be split into separate classes. An attribute-hiding factor of less than one (100%) indicates the use of public data and a corresponding lack of encapsulation.

Metrics such as depth of inheritance (MOOSE\_DIT), number of ancestors (QMOOD\_NOA) and number of children (MOOSE\_NOC) can highlight areas of the design which, whilst not incorrect, are more complex and worthy of additional test effort. A high depth of inheritance or number of ancestors indicates a class which supports many interfaces, which must be tested thoroughly to show that it correctly implements each one. A high number of children or ancestors indicates a class whose design is crucial to the system, for which changes are likely to be expensive.

Other metrics, such as the methods available (METH\_ALL), response for a class (MOOSE\_RFC), coupling between objects (MOOSE\_CBO) and method hiding factor (MOOD\_MHF95) can be used to estimate test effort. The number of



**Estimated testing effort**

methods available in the class (which includes inherited methods) and the response for a class indicate the number of distinct ways in which the class can respond to messages, and is clearly related to the number of test cases required for a thorough test. A high level of coupling between objects occurs when objects interact with many other objects, making the setting-up of a suitable test environment more complicated. The method-hiding factor (the proportion of methods which are declared private) indicates how much of the functionality of the class cannot be invoked directly from a black-box test, forcing the use of indirect test techniques.

By using Cantata++ to calculate and analyse these metrics developers and project managers can focus their testing efforts where they are most effective, helping them to produce higher quality, more-reliable software.

## Recent Ports

Cantata++ has recently been ported to a large number of new environments and compilers. The main ones include Solaris 7, HP-UX 11, Red Hat Linux, and LynxOS running on a PowerPC. Cantata has been ported to the C166 target running the Nucleus RTOS.

AdaTEST 95 has been ported to the Aonix/Raven environment (Intel x86 target).

Readers are reminded that all current IPL product platform availabilities are posted on our website, but that if you require support for a platform not posted, please discuss the matter with yoursupplier.

## Compilers Have Faults Too

A plea from the Tech Support room: readers and users are reminded that compilers are not infallible pieces of software. This seems to apply most particularly to C++ compilers. In fact, in a surprisingly high proportion of problems reported to the IPL Cantata++ support team the cause has been found to lie with the compiler. The principal difficulties have been with compilers that reject legal ISO C++ code from Cantata++, and compilers that accept illegal ISO C++ user code that was then rejected by Cantata++. These situations are frequently difficult to analyse and users are urged to be patient. Just because the problem is shown up by the test tool please do not assume that the testtool is at fault!

## AdaTEST Audit

We are pleased to announce that AdaTEST has again been successfully audited for qualification on a DO-178B Level A project. This latest one was by Meggitt Avionics for use on their SFD Mk 2 project for Boeing Aircraft. The project environment was Aonix ObjectAda on Solaris for host testing, and the target was TI C40 programmed with the Tartan compiler.

## Testing Ted



Gilchrist & Downing

## Support

Don't forget to keep your eye on our product support area at our web site: ([www.iplbath.com/prodsupp/](http://www.iplbath.com/prodsupp/)) for news of product upgrades and tips.

## AdaTEST/95

### Updates

At long last the new versions of AdaTEST and AdaTEST 95 have been completed. These will NOT be shipped automatically to customers under maintenance. If you would like to know more please contact Ian Gilchrist or yoursupplier.

## News about News

For those who are unable to wait six months between the successive issues of Testing Times IPL is now publishing a monthly digest of product news on its website. Check it out at <http://www.iplbath.com/p40.htm>.

## Meet Us

- ▶ IPL will be exhibiting at SQEEurope in London, 10 to 12 May 2000.
- ▶ We will be taking part in two 'Safety and Mission Critical Software' seminars being hosted by Aonix:
  - May 16, near Reading UK
  - May 18, near Derby UK
- ▶ IPL will be exhibiting at the Ada Europe conference being held this year at Potsdam Germany, June 27-29.

Contact Ian Gilchrist for details of any of these events.

## IPL Overseas - New Agency Agreements

IPL has forged several agency agreements with companies in Sweden, India, Benelux and Denmark. These companies are Nohau Elektronik in Malmo, RAMP Technologies in Bangalore, TESS in Markelo (Netherlands), and Nohau in Copenhagen. It is hoped that over time these relationships will develop into full distributor relationships.

## What Millennium Bug?

IPL has not heard of any errors or faults in our products which might be attributed to the Y2K bug. If you have experienced or heard of any, we would be interested to know of them. Contact Ian Gilchrist.

## Training Dates

Product training courses are intended to teach new users enough about Cantata, Cantata++ and AdaTEST to become productive with the tools. The following are the currently scheduled dates for courses in Bath:

### Introduction to Cantata:

2000: May 16, July 18, September 26, November 21.

### Introduction to Testing C++ with Cantata++\*:

2000: May 17, July 19, September 27, November 22.

### Using Cantata++\*:

2000: May 18, July 20, September 28, November 23.

### Introduction to AdaTEST:

2000: May 19, July 21, September 29, November 24.

\* These courses are also available as multi-media CD-ROMs.

The above courses are of one day duration and are presented at our offices. If you are interested in any of the above, AdaTEST 95 training or on-site training courses, please contact us for further details. Please also see: <http://www.iplbath.com/p72.htm> for further training course information.

## Further Information

An information request sheet is enclosed with this newsletter. Also remember that most of our literature is available for immediate access at the web address given below.

**IPL Software Products Group,**  
 Eveleigh House, Grove Street, Bath,  
 BA1 5LR, UK  
 Tel: +44 (0)1225 475000  
 Fax: +44 (0)1225 444400  
 email: [tools@iplbath.com](mailto:tools@iplbath.com)  
 WWW: <http://www.iplbath.com/tools>

### German Office:

**IPL Software Products Group, Zugspitzstr. 2,**  
 D - 85591 Vaterstetten, Deutschland.  
 Phone: +49-8106-4034  
 Fax: +49-8106-4035  
 email: [germany@iplbath.com](mailto:germany@iplbath.com)

## Distributors

**North America: Quality Checked Software**  
 Contact: Scott Thomas, [cst@qcsld.com](mailto:cst@qcsld.com)  
 Tel: +1 503 646 9991, Fax: +1 503 646 8783

### Italy: Instrumatic 2000

Contact: Salvatore De Luca, [sdl@instrumatic.it](mailto:sdl@instrumatic.it)  
 Tel: +39 02 9379 9321, Fax: +39 02 9379 9195

