Tony Elliston

SIGADA 2003
Model Based Real-Time Software Design

tools for mission critical software development
About the TNI Group

- Aerospace
- Industrial Automation
- Telecom
- Semiconductor
- Automotive

Software Engineering
Hardware design
Requirement traceability
Formal validation

70 Engineers
Software Design

- Software Requirements Analysis
- Software Design
- Software Coding
- Software Unit Testing
- Software Functional Testing

System Requirements
- Hardware
- Operating System

Realization Constraints
- Team Organization
- Technical Choices
- Quality Standards

Company Confidential
SOFTWARE PRODUCTS

CP HOOD
STOOD
REQTIFY
ORCHIS
COSIMATE
CONTROLBUILD
MAJOR PROJECTS USING CP HOOD?

EUROFIGHTER  TYPHOON

TORNADO

NIMROD

HAWK

HARRIER

AM346
Reqtify

A light and powerful solution for requirements traceability
Why use Requirements Traceability?

31% of all software projects are canceled before completed, 50% of projects cost over 190% of the original estimate, 9% on time and on budget (large companies) 16% on time and on budget (small companies),

Source: “CHAOS Surveys and Reports” - The Standish Group - www.standishgroup.com
Reqtify : Easy to integrate

A non-intrusive approach :
No modification of your development and configuration management process.

Traceability during the whole process
(text tools, analysis and modelling tools, code, …)

Qualified DO178-B as a verification tool for A380,
complies with D0254 and other standards.

Simple user interface allowing powerful
navigation in the traceability graph

Reqtify can be used even on projects
already started !
Reqtify : Immediate ROI

A minimal investment:

- Easy to handle, very short training course,
- No need for database administration,
- A floating licence,
- Windows / Unix interoperability

A small investment in Reqtify and training can provide a truly extraordinary payback even on the first project.
The HOOD design process,
( + semantics, graphical &
textual notations )

The AADL semantics
( + textual notation )

The UML 2.0 standard
graphical notation
“Good” Software Engineering practices:
- well defined architectural & detailed design process (HOOD)
- model-based engineering (designing before coding)
- requirements traceability management
- documentation framework
- complying with standards: DO-178; ECSS-E40; AADL; UML 2.0;

Managing the complexity:
- designing in the large:
  - hierarchical decomposition
  - components management
- distributed development:
  - modularity - interfaces
  - multi-users
- mixed paradigms:
  - function oriented
  - object oriented
  - task oriented

Modeling real-time:
- reaching deadlines:
  - mastering control flows
- no loss of data:
  - mastering data flows
- predefined HRT components
  - cyclic - sporadic - protected
- supporting RT executives
Stood Features Summary (2)

Supported platform:
- Solaris, Hpxx, Aix, Linux, Windows
- Full Unix/Windows interoperability

Versions:
- Current version: v 4.3
- End 2003: v 5.0
- Full upwards compatibility

Documentation generators:
- PostScript
- HTML
- RTF (Word)
- MIF (FrameMaker)
- TPS (Interleaf)

Code generators:
- Ada 95
- Ada Ravenscar
- C with RTOS
- C++
- AADL

Code reverse engineering:
- Ada
- C
- AADL

Interchange:
- SIF (CP HOOD)
- XML/CASEML
STOOD Release 5

- UML 2.0 analysis front end
- New advanced GUI
- Support for AADL
Who uses STOOD/ Reqtify?

- **AIRBUS** for A340 and A380 software and avionics
  Corporate agreement including subcontractors

- **THALES**
  Corporate agreement including subcontractors

- **MBDA** for missile software developments

- **ALCATEL Space** for Satellite ground projects

- **EUROCOPTER** for the TIGER helicopter

- **CNES** (French space agency) for Satellite projects
Thank you