Creating Componentised Systems for EW

Neil Brearley, Leonardo Airborne and Space Systems, Luton
Introduction and Contents

• Context - Defensive Aid Systems (DAS) and Electronic Warfare (EW)
• System Complexity
• The classic software upgrade cycle – what consumes time and effort?
• Changing the Paradigm - insertion of ‘modular components’ to a fixed core system – the ‘EW App’
• Use MathWorks® tools to generate Apps for Typhoon DAS
• What are the benefits of this Approach?
• Realising the dream – the challenges of certification and re-qualification

• Questions
What does a Defensive Aids system do?

Even sophisticated modern jets face a threat from missiles:
- They come in many shapes and sizes
- Ground batteries often have mobile launch vehicles
- Tracking radars are very capable and use latest electronics
- Some have engagement ranges out to several 100 kms
- They are subject to constant upgrade and change

A Defensive Aids system performs the following tasks:
- Detect and identify threats at long range
- ‘Situational Awareness’ - warn the pilot of impending danger
- Monitor threats for evidence they are tracking you
- Assign and deploy countermeasures to evade tracking
- Detection of missile launch
- ‘Last ditch survival’ - missile evasion and counters
Typhoon DASS: A complex system!

‘Brains’ of the system and the host for ‘Apps’
DASS system hardware on the development rig
## Software Update ‘Spiral’ – Where does the effort go?

<table>
<thead>
<tr>
<th>Supplier Steps</th>
<th>Prime and RAF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement ‘negotiations’</td>
<td>Avionics Integration test</td>
</tr>
<tr>
<td>Design and code changes</td>
<td>End user ground trial</td>
</tr>
<tr>
<td>Static testing</td>
<td>Evaluation Flight Trials</td>
</tr>
<tr>
<td>Integration Rig test</td>
<td>Fleet Embodiment</td>
</tr>
<tr>
<td><strong>Formal test and Issue</strong></td>
<td>Test and re-qualification has become the dominant factor in time and cost</td>
</tr>
</tbody>
</table>
Changing the Paradigm – modular insertion

"App Manager" acts as a 'broker' to trigger then interface to an appropriate App. It is fully qualified as part of the 'base system'.

The base system must have a 'default' behaviour if any App fails to respond as monitored by the App Manager.

This could be a logical or a physical barrier - in our case it is both.
Using the MathWorks Toolset to Create Apps

Leonardo has used MATLAB and Simulink for several years to model system behaviour and to prototype algorithms. It is a natural extension to create an executable that can be configured as a loadable App. The ‘c’ auto-coder is used to produce an executable.
Development environment advantages

- Ease of using MATLAB error reporting and debug tools
- Run-time profiler very useful for benchmarking and to focus optimisation work
- Shared library for threat simulation and receiver characteristics
- Able to set-up ‘c’ auto-code options for host and target compatibility
- Can control auto-code to forbid dynamic memory and perform useful error trapping
- Recommend bringing in expert help when getting started e.g. to configure the development environment
Benefits of the Modular Approach

1. De-coupling of development activities
   - Allows additional components to be designed and developed independently
   - Makes better use of limited rig availability

2. De-coupling of Test and Qualification activities
   - Base-line system qualified once
   - Avoids re-testing functions that have not changed
   - Faster update cycle for urgent changes (new threat etc.)

3. Modelling and prototyping environment
   - Direct relationship between modelled output and target executable
   - Facilitates standard function libraries for both models and algorithms
Realising the dream

There are several hurdles to be overcome in order to realise the App concept operationally:

1. Proving the ‘fire-wall’ and recovery in the event of App failure
2. App performance and overheads of the App Manager interface
3. Accreditation of the development route and associated tool-set (compiler etc.)
4. Aircraft-fleet configuration management
5. Security aspects
6. Prime contractor and end-user (RAF trials) test burden
Any Questions?
THANK YOU FOR YOUR ATTENTION