



*Lessons Learned Using COTS in
Satellite Command & Control*

Ground System Architecture Workshop

February 1997

Why COTS?

- *Systems based upon COTS products are significantly less expensive to procure & maintain*
- *COTS products guarantee technological growth*
- *Intelligent COTS product selection and integration offer more flexibility than custom systems*
- *COTS products greatly reduce development timelines*
- *Systems utilizing significant COTS content have proven highly successful*
 - *CERES / RSC*
 - *Inmarsat*
 - *Hughes OS/702*

COTS Applications

- *COTS products are now available to support the full range of of satellite command and control functions*
 - *Telemetry acquisition and processing*
 - *Commanding*
 - *Data display and visualization*
 - *Data distribution / middleware*
 - *Orbit data processing*
 - *Flight dynamics*
 - *Mission scheduling & planning*
 - *Simulation*

COTS Means New Methodologies

- *Traditional development practices, e.g. 2167A, do not readily apply to COTS product integration*
- *Upgrade policies must be considered early*
- *The role of a systems integrator is critical to COTS system*
- *Configuration management methodologies must be altered to accommodate COTS structures*
- *COTS systems are dynamic and long term logistics must be designed to support layered upgrades*

Picking COTS

- *Functionality of offered product*
- *How can the product be customized by the integrator*
- *How well supported; responsiveness of vendor*
- *Frequency of upgrades - how often and how much*
- *Documented interfaces to support integration*
- *Install base - how many and for what use*
- *Platforms supported*
- *Available training programs*
- *Make sure its true COTS - not repackaged legacy code*

COTS versus NOTS



	<u>COTS</u>	<u>Custom</u>
Developed using private funds	X	
Rehosted government SW or IR&D funded		X
Published price list (uniform pricing)	X	
Pricing handled on a case by case basis		X
On-going, dedicated development staff	X	
SW available on multiple, standard platforms	X	
Standard maintenance policy	X	
Maintenance available by quote only		X
Purchased by wide customer base	X	
Update versions produced regularly	X	
Documentation consistent with commercial standards	X	



How to Integrate?

- *Software integration methodologies are driven by long term requirements*
 - *Point to point interfaces*
 - *Use of middleware*
- *Middleware utilization is optimal for:*
 - *Large number of COTS products*
 - *Dynamic system configuration*
 - *Requirement for product substitution*
 - *Highly object oriented designs*
- *Middleware adds extra product costs but can reduce long term maintenance costs*
- *Middleware can add additional reliability*

Issues with COTS

- *Concern over the future of any specific product or its vendor*
- *Multiple COTS product updates require integration and installation coordination*
- *Multiple COTS products can mean multiple GUIs*
- *As with most custom systems, rapid advances in hardware technologies can quickly outdate some COTS hardware components*
- *Users may have less control over changes in COTS-based systems than in custom systems*

The Future is COTS

- *COTS provides thoroughly tested, proven capabilities*
 - *Large installation base demands thorough testing*
 - *User feedback and user groups promote well-designed, complete products*
- *COTS allows simplified, flexible system integration*
 - *COTS products support open architecture concepts and standards*
 - *APIs, IPCs, bridges, etc. reduce integration efforts*
 - *Easy replacement of COTS components with comparable products*
- *COTS has proven to be economical*
 - *Product development costs spread across full installation base*
 - *Use of COTS eliminates the need for costly maintenance and management of custom source code*
- *Profitable organizations have embraced COTS*

