Modernizing Tomahawk on the Cheap

Robin Ross
NSWC Dahlgren Division
TTWCS SDA Software Lead
Work: 540.653.2109
robin.ross@navy.mil

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.
Agenda

- Background
- Problems faced
- Alternatives Assessed
- Solution chosen
  - Hardware Architecture
  - Porting Considerations
- Results
**Background**

- Tomahawk Weapon Control System (TWCS) is mature
  - Evolved from TWCS to TTWCS v5.3 over 30 years
  - TWCS developed in 80’s
    - Proprietary Mil-Spec computers and Operating System (OS)
  - Advanced Tomahawk Weapon Control System (ATWCS) developed in early 90’s
    - Commercial Off-The-Shelf (COTS) computers
    - Proprietary OS
  - Tactical Tomahawk Weapons Control System (TTWCS) developed in late 90’s
    - COTS VME-based computers
    - Proprietary OS
- Multiple funding requests for modernization were unfunded

**Flowchart:**

1. TWCS (v1) → ATWCS TCGR (v2) → ATWCS LCGR (v3) → TTWCS (v4) → TTWCS (v5.3) → TTWCS (v5.4)
Problems Faced

● Computer Resource Constraints
  – Deployed versions of TTWCS constrained by processor memory and CPU speed

● Viability Concerns
  – Unsupported COTS software (SW)
  – Obsolete hardware (HW)
  – DoD mandates for Open Architecture and Security

● Budgetary Concerns
  – DoD dollars increasingly difficult to obtain

Needed an affordable, step-wise approach that would be the foundation for future modernization
Alternatives Assessed

- Add additional processor in existing racks
  - Minor impact to ship configuration; no change to footprint
  - Low risk
  - Least expensive

- Replace all processors
  - Major ship configuration impact
  - High risk
  - High cost

- Replace all processors, network, and displays
  - Complete ship configuration impact
  - Even higher risk
  - Most expensive
Solution Chosen

- First step in incremental approach to modernization of Tomahawk Weapon Control System
  - Insert x86 processor into each Tomahawk equipment rack
  - Use Linux OS on new processor
  - Port selected pieces of SW to new processors
- Future increments will remove HP processors and port remainder of code
Porting Considerations

- Careful consideration given to which SW components were ported to new x86 processors
  - Safety Critical components not ported to reduce risk to safety certification of build
  - External Interfaces
    - Endian sensitive interfaces
      - Legacy HP processors are big-endian
      - X86 processors are little-endian
      - Point-to-point interfaces expected big-endian format
      - Components with these interfaces were not ported since each message would require complex byte and bit swapping due to bit fields crossing byte boundaries
    - Components with Common Object Request Broker Architecture (CORBA) interfaces were good candidates
      - Built-in data marshalling eliminates need for byte swapping
Porting Considerations (cont)

– Multi-core sensitivity
  - Ada components not designed for multi-core processors
    - Work required to counter the effects of running on multi-cores

– Resource usage
  - Highly algorithmic components work well with multi-cores
  - CPU-intensive components benefit from speed of new processors
  - SW components with larger memory demands benefit from additional memory
Results

- ~50% of code ported to x86 processors
- 15 months initial development to enter system test
- Significant performance enhancements seen during system test
  - Functions that took minutes now take seconds
- Certifications effort streamlined due to focused areas of impact
- Operational Test (OT) extremely successful
  - Deemed effective and suitable
  - "The system met all critical effectiveness performance parameters with noticeably improved processing times for key functions." (RADM Dunaway’s IT-CF OT Commander’s Report November 15, 2011)
- Fleet Release decision reached December 2011