Monitor & Control System (MCS)
Critical Design Review

Steve Ellingson on behalf of the VT LWA group
Virginia Polytechnic Institute & State University
ellingson@vt.edu

November 20, 2009
Virginia Tech’s Role in LWA

• VT subcontract accounts for about 6% of ONR LWA FY06-FY08 funding

• Current Responsibilities in LWA Project:
  – Monitoring & Control System (MCS) – about 2/3 of effort, beginning Fall 2008
  – MCS Data Recorder (MCS-DR)
  – Station-Level Calibration (SLC)

• Past / Recurring Efforts
  – Systems Architecture / Systems Engineering
  – Rapid Prototyping / Equipment Loan: ARX, Digitizer, S60 data recorder
  – Data Analysis, RFI, Transient Search Software

• All VT deliverables (including software) are freely available at http://www.ece.vt.edu/swe/lwavt/
Virginia Tech People Involved in MCS

• Steve Ellingson
• Cameron Patterson
• John Simonetti

• Current Students:
  – Mahmud Harun
  – Qian Liu
  – Chris Wolfe
  – Abirami Srinivasan
  – Sushrutha Vigraham

• Past Students:
  – S.M.S. Hasan
  – Kyehun Lee
  – Wyatt Taylor
MCS Deliverables Most Relevant to CDR

• MCS Subsystem Description

• MCS Interface Control Document (ICD)

• MCS Software

• MCS Data Recorder (MCS-DR)
             Aug 26, 2009. Also available as LWA Memo 165.
             Data Recorder,” Oct 10, 2009

• System Diagnostic & Emulation Software
  [MCS0015] A. Srinivasan and S. Ellingson, “Python code for direct communication with subsystems,”
             Aug 7, 2009
What is MCS?

• Defined in MCS0004, MCS0007

• **5 parts:** Scheduler, Executive, Task Manager, Data Recorder (DR)

• **6 functions:**
  
  – Control → Scheduler
  
  – Monitoring → Scheduler, Executive
  
  – Logging → Executive
  
  – User Interface → Scheduler, Task Manager
  
  – Some application software → Task Manager
  
  – Interim data recording system → DR
    
    • in lieu of Data Aggregation & Comm (“DAC”) subsystem

• **Not** Data Reduction (but...)

• **Not** Data (Re)formatting (but...)
MCS Architecture

- MCS0007
- MCS communicates with subsystems via UDP-based MCS Common ICD protocol (MCS0005)
- There is no physical “console”
  - Communication with MCS is through Gateway using standard internet protocols (primarily ssh)
- Changes since PDR:
  - TCD physical interfaces to MCS/Scheduler removed (NTP is sufficient)
  - Tape storage replaced with “Data Recorder Storage Units” (DRSUs), described later
MCS Readiness Summary

- MCS Hardware Design [MCS0023]
  - Complete

- MCS/Scheduler Software Design [MCS0021]
  - Software is available in a functional pre-alpha release status
  - SHL and ASP are fully supported; DP partially supported
  - Scheduling implemented but not yet supported through interface
  - 90% complete; risk associated with completion is very low

- MCS/Executive & MCS/Task Manager Software Design
  - 10% complete; risk associated with completion is low.
  - Risk further offset by fact that station can be controlled via a bash-scriptable command line interface using only MCS/Scheduler (see example later) [MCS0021]
MCS CDR Readiness Summary

- MCS Performance
  - Most demanding requirement is the ability to re-point a “calibration beam” within 5 ms, repeating at every 60 ms (LWA Memo 146).
  - We've confirmed that MCS will be at least 2 orders of magnitude faster than necessary, and is limited primarily by network throughput.

- MCS size, power, and cost: [MCS0023]

  *MCS not including Data Recording (MCS-DR):*

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Power</th>
<th>Cost</th>
<th>Status (Nov 10, 2009)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduler</td>
<td>computer[^1,^2]</td>
<td>750W</td>
<td>$1916</td>
<td>Purchased; at VT</td>
</tr>
<tr>
<td>Executive</td>
<td>computer[^1,^2]</td>
<td>750W</td>
<td>$1916</td>
<td>Not yet ordered</td>
</tr>
<tr>
<td>Task Processor</td>
<td>computer[^1,^2]</td>
<td>750W</td>
<td>$1916</td>
<td>Not yet ordered</td>
</tr>
<tr>
<td>Gateway</td>
<td>managed switch[^3]</td>
<td>50W</td>
<td>$1528</td>
<td>Purchased; at VT</td>
</tr>
<tr>
<td>Command Hub</td>
<td>managed switch[^3]</td>
<td>50W</td>
<td>$1528</td>
<td>Not yet ordered</td>
</tr>
<tr>
<td>Other[^4]</td>
<td>(misc)</td>
<td></td>
<td>$1000</td>
<td>2U</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td><strong>2350W</strong></td>
<td><strong>$9804</strong></td>
<td>10U</td>
</tr>
</tbody>
</table>

*From PDR [MCS0008]: 2000W $10000 24U*
MCS-DR CDR Readiness Summary

- MCS-DR Hardware Design  [MCS0019, MCS0020, MCS0023]
  - Complete for DRs #1 and #2
  - New PCs required for DRs #3, #4, #5 – Selected PC was discontinued by vendor
  - Moderate risk that new PC will not work. Physically larger PC might be required

- MCS-DR Software Design  [MCS0022]
  - Proof of design achieved (see below)
  - Greater than 90% complete; remaining effort is MCS integration
  - Moderate risk that substantial changes will be required for new PCs

- MCS-DR Performance  [MCS0018]
  - Demonstrated 10 hours of continuous error-free acquisition at 115 MB/s, repeatable
  - Exceeds requirement of 112 MB/s (corresponding to TBN at its highest bandwidth)
  - Tests done using a second computer emulating DP, since DP is not yet available

- MCS-DR #1 is installed in the shelter & operational
# MCS-DR CDR Readiness Summary

**MCS-DR:** [MCS0023]

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Power (W)</th>
<th>Cost ($)</th>
<th>Status (Nov 10, 2009)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Recorder PC#1</td>
<td>computer ¹</td>
<td>360W</td>
<td>$1799</td>
<td>4U</td>
</tr>
<tr>
<td>Data Recorder PC#2</td>
<td>computer ¹</td>
<td>360W</td>
<td>$1799</td>
<td>4U</td>
</tr>
<tr>
<td>Data Recorder PC#3</td>
<td>computer ²</td>
<td>350W</td>
<td>$1989</td>
<td>4U</td>
</tr>
<tr>
<td>Data Recorder PC#4</td>
<td>computer ²</td>
<td>350W</td>
<td>$1989</td>
<td>4U</td>
</tr>
<tr>
<td>Data Recorder PC#5</td>
<td>computer ²</td>
<td>350W</td>
<td>$1989</td>
<td>4U</td>
</tr>
<tr>
<td>10GbE NICs ³ (2 @ $595 each)</td>
<td></td>
<td></td>
<td>$1190</td>
<td>2U</td>
</tr>
<tr>
<td>(3 @ $595 each)</td>
<td></td>
<td></td>
<td>$1785</td>
<td>3U</td>
</tr>
<tr>
<td>1GbE NICs ⁸ (5 @ $85.99 each)</td>
<td></td>
<td></td>
<td>$430</td>
<td>5U</td>
</tr>
<tr>
<td>Video cards ⁹ (2 @ $135.99 each)</td>
<td></td>
<td></td>
<td>$272</td>
<td>7U</td>
</tr>
<tr>
<td>Video cards ¹⁰ (3 needed, $135.99)</td>
<td></td>
<td></td>
<td>$408</td>
<td>9U</td>
</tr>
<tr>
<td>CXP4 cables (5 @ $105 each)</td>
<td></td>
<td></td>
<td>$525</td>
<td>11U</td>
</tr>
<tr>
<td>DRSU x 5</td>
<td>data storage ⁴</td>
<td>500W</td>
<td>$1750</td>
<td>2U</td>
</tr>
<tr>
<td></td>
<td></td>
<td>750W</td>
<td>$2625</td>
<td>3U</td>
</tr>
<tr>
<td>Other ⁷</td>
<td>(misc)</td>
<td></td>
<td>$1000</td>
<td>4U</td>
</tr>
</tbody>
</table>

**TOTAL** | | 3020W | $19550 | 29U |

From PDR [MCS0008]: 3500W $15500 36U not including mass storage (tapes) additional for tape drives
MCS-DR

View from Front

- This is one of 5 MCS-DRs
- PC: Dell Studio XPS 435MT (Intel Core i7-940, 2.93 GHz), 6GB Tri-Channel 1066 MHz DDR3 SDRAM, 1TB HDD, Fitted with Myricom 10GbE NIC
- Replacement: Dell Precision T1500 (Intel Core i7-870, 2.93 GHz), 8GB Dual-Channel 1333 MHz DDR3 SDRAM, 80 GB HDD
- Cable between MCS-DR PC and DRSU is 1 m

View from Rear
- $875 each
- 5 TB (at least 10 hours at highest possible rate)
- 1U rackmount, easy to move
- 2 connections (AC power, eSATA)
- Convenient for long term storage?
MCS-DR Software Architecture

OS: Ubuntu Linux 9.04, 64-bit
App: ANSI C using POSIX “librt” library for asynchronous I/O
MCS/Scheduler

- Dell Precision Workstation R5400
  - 2U rack mount
  - Quad Core Intel Xeon E5405 (2.0 GHz)
  - 4 GB RAM, 250 GB HDD, Dual 1GbE
- Ubuntu Linux 9.04, 64-bit

- Software in ANSI C  [MCS0021]
  - “ms_init” process initializes, launches processes, then terminates
  - One “ms_mcic” process per LWA level-1 subsystem
  - “ms_exec” is main loop, communicates with subsystems through the ms_mcic processes via persistent POSIX message queues
  - ms_mcic processes maintain subsystem state (MIBs) in dbm files
MCS “User-Side” Interfaces

- **Tier 0**: MCS Common ICD [MCS0021]
  - No MCS software used (doesn't even need MCS/Scheduler)
  - Intended primarily for subsystem development and integration
  - Some software provided for this (MCS0013, MCS0015)

- **Tier 1**: MCS/Scheduler Linux command line interface (see next slide)
  - “msei” to issue commands, “ms_mdr” to read MIBs
  - Bash scripts can be used to run station using only this much
  - Current release is 0.4 (MCS0021)

- **Tiers 2 and higher** involve MCS/Executive and applications running on MCS/Task Processor
  - Lowest tier accessible by a user can be administered through login privileges; should depend on the expertise of the user and level of trust in the user
## Tier 1 Interface (MCS/Sch Command Line)

### example_init.dat

```
mibinit SHL 192.168.10.2 1738 1739
mcic    SHL
mibinit ASP 192.168.10.3 1740 1741
mcic    ASP
mibinit DP_ 192.168.10.4 1742 1743
mcic    DP_
```

### command line examples

```
$ ./ms_init example_init.dat

$ ./msei SHL INI '00090&2.5&100000'
$ ./msei SHL RPT PORTS-AVAILABLE-R1
$ ./msei SHL PWR '104ON '

$ ./msei ASP INI 16
$ ./msei ASP RPT ARXSUPPLY-NO
$ ./msei ASP FIL '02702'

$ ./msei DP_ RPT LASTLOG
$ ./msei DP_ TBW 1 0 36000000
$ ./msei DP_ TBN 38.0 7 15 0

$ ./ms_mdr SHL SUMMARY
NORMAL
090825 18:03:04

$ ./msei MCS SHT
```
MCS/Executive & MCS/Task Manager

• These use same PC as MCS/Scheduler

• MCS/Executive
  – Parses observation requests into MCS/Scheduler commands
  – Performs periodic calibration, diagnostics, and safety monitoring
  – Manages station MIB

• MCS/Task Manager serves “user apps” (command line and GUI versions):
  – Monitoring Application
  – Scheduling Application
  – Frequency-Domain Analyzer (FFT/Spectrum Analyzer)
  – Time-Domain Analyzer (time scale/resolution from Nyquist to days)
  – Time-Frequency Domain Analyzer (Spectrograms)
  – Sky Map ($T_B$ from model, ephemeris for a few key sources)
  – Command Line Help Application
MCS Testing

• We have developed software that emulates SHL, ASP, DP, and an arbitrary number of “generic” MCS Common ICD-compliant subsystems.

• Any of these subsystems can be simulated using a PC running the emulation software.

• MCS testing is facilitated by a small LAN — identical to the one in the shelter — consisting of actual MCS computers and simulated subsystems.
  – This provides a very compelling workout for MCS, but does not completely remove the risk that unexpected subsystem behavior will cause problems.

• All MCS software and emulators can also run, without modification, all on one PC (using local loopback IP in lieu of a physical LAN).
  – This is useful for development and demonstration.
Summary & Path Forward

- Primary risks (MCS-DR transfer rate, MCS/Scheduler basic function) have been mitigated.

- On track to meet the LWA-1 system readiness milestones promulgated by the Project Office after PDR; in particular, the December 2010 “Full Station” milestone:
  - The primary risk to this is not receiving a NCTE to the current March 31, 2009 expiration of funding.
  - Moderate risk that replacement for MCS-DR PCs 3-5 is not suitable.

- Top-level tasks going forward:
  - Validate the new MCS-DR PC
  - Complete MCS/Scheduler software, including support for DP and MCS-DR
  - Complete MCS/Executive software
  - Complete MCS/Task Processor software
Closing Remarks

• DP ICD has not been updated since PDR. Necessary changes identified since PDR have been assumed to be adopted in the development of MCS software

• Subsystem developers should use Tier 0 and Tier 1 software to develop and test their MCS interfaces. This will potentially save a lot of grief when time comes to integrate subsystems, or implement new capabilities

• Strongly recommend thorough testing of the DP to MCS-DR interface before trying this in the shelter. (Remember, MCS-DR was verified using our simulation of DP output)

• Transport / archiving / management of user data has the potential to become complicated and onerous. Advance planning is suggested