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

[MCS0004] S. Ellingson, "MCS Subsystem Definition," Ver. 2,Feb. 23, 2009 [MCS0007] S. Ellingson, "MCS Architecture," Ver. 4, Nov 7, 2009 [MCS0023] S. Ellingson, "MCS Documentation for LWA-1 CDR," Nov 10, 2009

MCS Interface Control Document (ICD)

[MCS0005] S. Ellingson, "MCS Common ICD," Ver. 1.0, April 4, 2009

MCS Software

[MCS0021] S. Ellingson, "MCS/Scheduler Software Version 0.4 (pre-alpha)," Nov 7, 2009.

• MCS Data Recorder (MCS-DR)

[MCS0018] C. Wolfe, S. Ellingson & C. Patterson, "MCS Data Recorder Preliminary Design & Verification," Aug 26, 2009. Also available as LWA Memo 165.

- [MCS0019] C. Wolfe, S. Ellingson & C. Patterson, "MCS-DR Storage Unit," Sep 23, 2009
- [MCS0020] C. Wolfe, S. Ellingson & C. Patterson, "Interface Control Document for Monitor and Control System Data Recorder," Oct 10, 2009

[MCS0022] C. Wolfe, "MCS Data Recorder Operating System Version 0.8 (pre-alpha)," MCS0022, Nov 10, 2009.

System Diagnostic & Emulation Software

[MCS0015] A. Srinivasan and S. Ellingson, "Python code for direct communication with subsystems," Aug 7, 2009

[MCS0013] A. Srinivasan and S. Ellingson, "MCS ICD Compliance Check Software," MCS0013, July 31, 2009

[MCS0012] A. Srinivasan and S. Ellingson, "MCS Common ICD Emulation Software for SHL," Aug 7, 2009

[MCS0009] S. Ellingson, "MCS Common ICD Network Check Software," MCS0009, Ver. 2, Apr 4, 2009



What is MCS?

- Defined in MCS0004, MCS0007 ٠
- **5 parts:** Scheduler, Executive, Task Manager, Data Recorder (DR) ۰
- 6 functions: ۲
 - Control
 - Monitoring
 - Logging
 - User Interface
 - Some application software
 - Interim data recording system \rightarrow DR

- \rightarrow Scheduler
- \rightarrow Scheduler, Executive
- \rightarrow Executive
- \rightarrow Scheduler, Task Manager
- \rightarrow Task Manager
- in lieu of Data Aggregation & Comm ("DAC") subsystem
- Not Data Reduction (but...)
- Not Data (Re)formatting (but...)

MCS Architecture

lèch

- 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 bashscriptable 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):

				,
Item	Description	Power Cost		<u>Status (Nov 10, 2009)</u>
Scheduler	computer ^{1,2}	750W \$1916	2U	Purchased; at VT
Executive	computer ^{1,2}	750W \$1916	2U	Not yet ordered
Task Processor	computer ^{1,2}	750W \$1916	2U	Not yet ordered
Gateway	managed switch ³	50W \$1528	1U	Purchased; at VT LWA1
Command Hub	managed switch ³	50W \$1528	1U	Not yet ordered
Other ⁴	(misc)	<u>\$1000</u>	2 U	
	TOTAL	2350W \$9804	10U	>

From PDR [MCS0008]: 2000W \$10000 24U

Nov 20 2009

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]

MCS-DR: [MCS	60023]				Nov 20, 2009
Item	Description	Power	<u>Cost</u>		<u>Status (Nov 10, 2009)</u>
Data Recorder P	C#1 computer ¹	360W	\$1799	4U	Purchased; at VT- LWA1
Data Recorder P	C#2 computer ¹	360W	\$1799	4U	Purchased; at VT Returning to VT
Data Recorder P	C#3 computer ²	350W	\$1989	4U	Ordered; awaiting shipment at VT
Data Recorder P	C#4 computer ²	350W	\$1989	4U	Not yet ordered
Data Recorder P	C#5 computer ²	350W	\$1989	4U	Not yet ordered
10 GbE NICs 3 (2)	2 @ \$595 each)		\$1190		2 installed (VT M&S)
(3	(a) \$595 each)		\$1785		3 ordered (VT M&S)
1GbE NICs ⁸ (5	@ \$85.99 each)		\$ 430		2 installed (VT M&S)
					3 on-hand (VT M&S)
Video cards ⁹ (2	@ \$135.99 each)		\$ 272		2 installed (VT M&S)
Video cards 10 (3 needed, \$135.99)			\$ 408		Not yet ordered
CXP4 cables (5	@ \$105 each)		\$ 525		Not yet ordered ⁶
DRSU x 5	data storage ⁴	500W	\$1750	2U	2 Purchased; at UNM ⁵ LWA1
		750W	\$2625	3U	3 not yet ordered
Other ⁷	(misc)		<u>\$1000</u>	4U	
	TOTAL	3020W	\$19550	29U	
	From PDR [MCS000	8]: 3500W	\$15500	36U	not including mass storage (tapes)
		175W	\$18000		additional for tape drives

MCS-DR

[MCS0020]

View from Front

View from Rear

- 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

DRSU [MCS0019]

- \$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 [MCS0018]

- 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 '1040N '
$ ./msei ASP INI 16
$ ./msei ASP RPT ARXSUPPLY-NO
$ ./msei ASP FIL '02702'
$ ./msei DP RPT LASTLOG
$ ./msei DP TBW 1 0 3600000
$ ./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, <u>without modification</u>, 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 <u>our simulation</u> of DP output)
- Transport / archiving / management of user data has the potential to become complicated and onerous. Advance planning is suggested

