Integrated Launch Operations Applications

Remote Display Developer

Cedric Flemming II

Kennedy Space Center NE-C1

BS Computer Engineering

KSC Intern - Spring 2014

2014/5/9

Integrated Launch Operations Applications

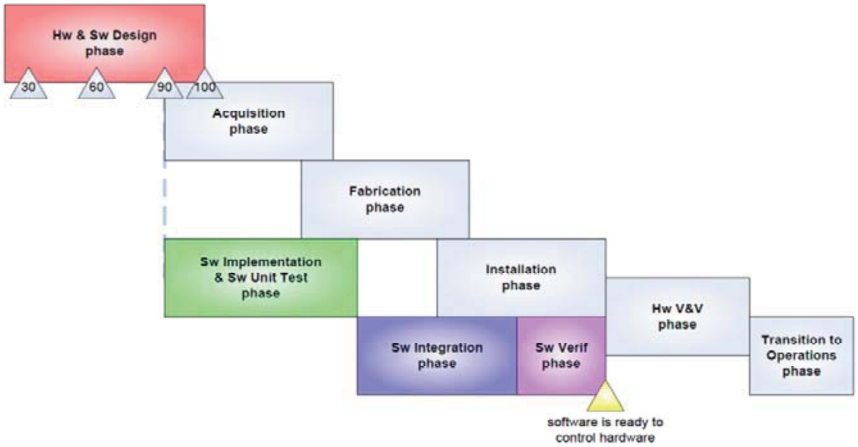
Remote Display Developer

Cedric M. Flemming II
Claflin University, Orangeburg, SC 29044

Abstract

This internship provides the opportunity to support the creation and use of Firing Room Displays and Firing Room Applications that use an abstraction layer called the Application Control Language (ACL). Required training included video watching, reading assignments, face-to-face instruction and job shadowing other Firing Room software developers as they completed their daily duties. During the training period various computer and access rights needed for creating the applications were obtained. The specific ground subsystems supported are the Cryogenics Subsystems, Liquid Hydrogen (LH2) and Liquid Oxygen (LO2). The cryogenics team is given the task of finding the best way to handle these very volatile liquids that are used to fuel the Space Launch System (SLS) and the Orion flight vehicles safely.

At the start of the internship, the design team was at 90% maturity of the requirements and design phase, getting applications ready for the development and unit test phase. As a team member, I support the Cryogenics team by following the Software Development Process (SDP) and meeting milestones and deliverables. Specifically, the support being provided is creating the LH2 and LO2 Firing Room applications and displays to be used in the Kennedy Space Center (KSC) Launch Control Center (LCC) during launch and offline processing, involves integrating the subsequent control applications that tie the application to a responding operation in the field, and in contributing a different perspective on topics discussed within the Cryogenics team.



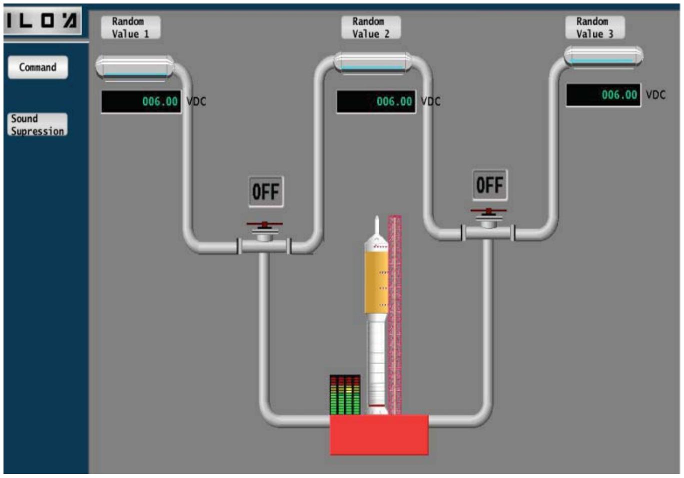
Nomenclature

COTS	Commercial off The Shelf
CUI	Compact Unique Identifier
DE	Display Editor
DVSApp	Display Verification Sheet Application
FR	Firing Room
JViews	A COTS product used to make displays
KSC	Kennedy Space Center
LCC	Launch Control Center
LCS	Launch Control System
ML	Mobile Launcher
MPS	Main Propulsions System
SCCS	Spaceport Command and Control System
SLS	Space Launch System
SRDS	Software Requirements and Design Specifications
TD	Test Driver

Introduction

After being officially trained and assigned to the Cryogenics system, I was given the task of developing the remote displays that will be used as part of the launch control system in the firing room at KSC. Cryos is responsible for the proper maintenance of Liquid Hydrogen (LH2) and LO2 (LOX) at Launch Pad 39B and Mobile Launcher (ML). Pad 39B and the ML will be used together for the future launch of the Space Launch System (SLS)/ Orion missions.

First lesson I received as a member of Cryos was that safety is of the most importance in the Cryogenics subsystem because "failure of any kind could result in loss of vehicle or loss of life, the Main Propulsions System (MPS) is often referred to as Main Problem System". The first task given to me was to create and edit several displays for Cryos and the MPS of the SLS Program. Initially I dealt with visual aspects of the 14 displays assigned to me by the team. A few displays had to be manifested from scratch in the LCS Display Editor, which is a customized version of IBM JViews. The displays had to have a certain look, layout and level of functionality to be used on the Launch Control Center (LCC) on launch day.



After the visuals were completed command and fusion Compact Unique Identifiers (CUI) were implemented, which are used to identify measurements and commands within the command and control. CUIs link the software to hardware and all the individual subsystems (Hypergolic, Ground Cooling System, etc.) software together.

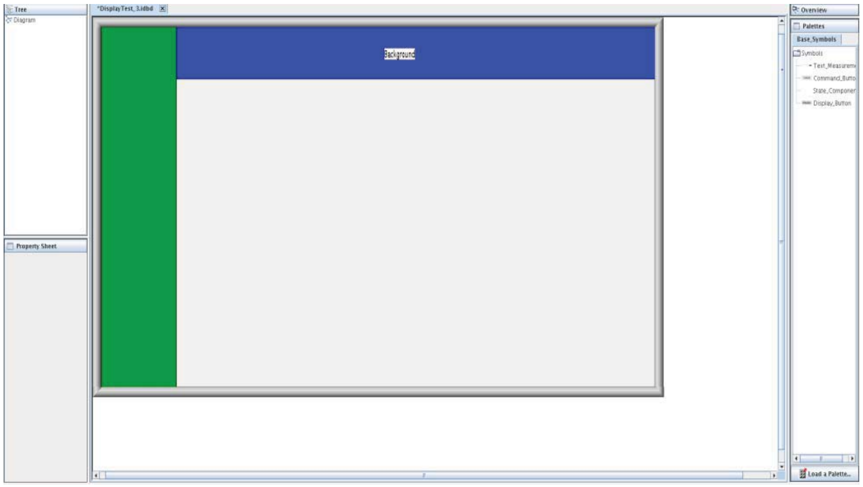
Example of a CUI: GMWDWD9VLVA126BY

All aspects of which are dictated by the Software Requirements and Design Specifications (SRDS) document.

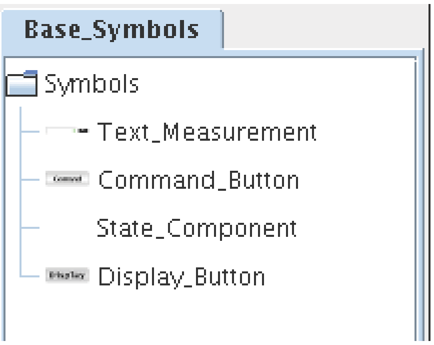
Display Developing

Typically, as previously stated, the making of a display is based on the SRDS. This will outline all the necessary components and visuals of a display, but a simple display will be made solely for the purpose of demonstration.

After opening the virtual machine, running the LCS display editor and choosing a destination to save it you are at the point where you can begin creating a display. The first thing that is placed in the display area is the background via the *icon* drawing primitive. The background is a predominantly a visual component, but has some functionality because it often shows which subdivision the display belongs to and it will contain all of the other components. The size of the background use can also vary depending on the purpose. Since this display is a demo it will not have a subdivision shown on it and the size used will be the default setting size. Once the preferred background is set the coordinates of the background must be adjusted in the *property sheet* located on the bottom-left of the display editor (DE). The preferred coordinates are (0, 0) because it places the background in the center of the display area. Selecting the grid icon under the view tab is also quite useful for aligning the background and all other objects on the display.



After the background is set display symbols, also called widgets, can be used to add to the display. The four symbols currently useable are:



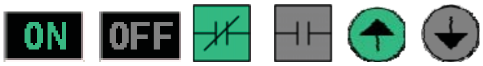
Text Measurement – That display numeric and non-numeric values like “on” and “off”. Text measurement are tied to a single CUI
Command Buttons – Issue a command, simply enough, and are tied to a single CUI
State Components – Use designated images to display an enumerated value and is tied to one CUI per symbol
Display Buttons – Can be used to connect to another CUI, no CUI involved

On the displays there are guidelines that can be used like:

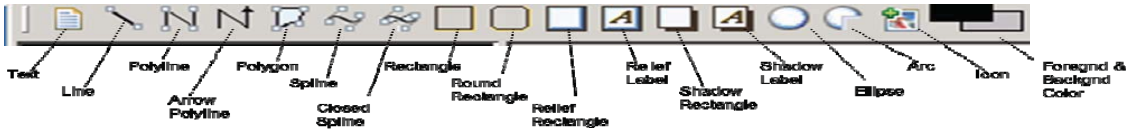
R	DEG R	temperature – degrees Rankin
Q	GPM	flow rate – gallons/minute
%	PCT	percent
V	VOLTS	voltage – volts
A	AMPS	current-amperes

There are also discrete data guidelines and graphics that are used:

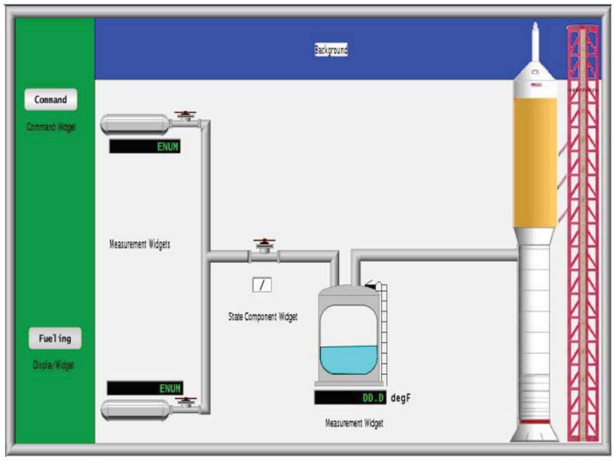
De-energized	Gray
Energized	Green
Bypassed	White
Transition	Magenta



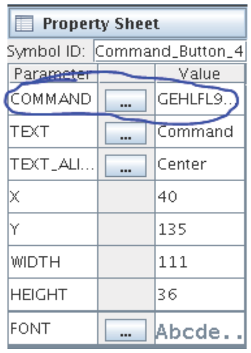
There are various methods to add to a display visually, which is provided by the Drawing Primitives. Thumbnail images have been used in JViews to give a better understanding of what symbols or shapes they make.



Using widgets and guidelines as well as numerous graphics that can used via the *icon* drawing primitive, the user can create a multitude displays. The display created for the demo shows the contents of two tanks need to be cooled then used to send to the rocket waiting for sound suppression. All of the piping, tanks, rocket and other none symbols were placed via the drawing primitive or the drawing tools at the top of the display editor. The Command Button on the left would do something along the lines of beginning the process of opening all valves to move fluids. The State Component would give you a visual representation to indicate rather or not the valves are open, closed or transitioning. Those images are populated in the "image thumbnail". The Text Measurements by the cylindrical tank indicated if they are closed (static) or open (flowing) and the Text Measurement under the large tank indicates the temperature.



The Display Widget on the bottom left is used to bring up another display that maybe directly affected by the currently used display. Once all of the necessary widgets have been placed into the display then the necessary CUIs have to be tied in. CUIs are selected from a bank that can be used on more than one display, tying them together. For State Components images have to be selected in the image thumbnail window to represent the different states that a particular component can be in at any given time. After the CUIs are assigned and the correct images are selected per the SRDS, the displays functionality can exercised in the display Test Driver (TD). If all display runs nominally in the Test Driver then it is ready for the next step in the development process, filling out a Display Verification Sheet Application (DVSApp).



Property Sheet

Symbol ID: Command_Button_4

Parameter		Value
COMMAND	...	GEHLFL9..
TEXT	...	Command
TEXT_ALI...	...	Center
X		40
Y		135
WIDTH		111
HEIGHT		36
FONT	...	Abcde..

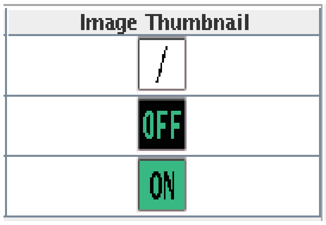


Image Thumbnail

Conclusion

Creating displays and tying in the corresponding CUIs are only a small portion of the total responsibilities of a Remote Display Developer. Displays have to go through several levels of verification and simulation before they are officially ready for use with the planned SLS/Orion missions. Throughout the entire process the developer has to be there to make sure everything is nominal. The tasks I have completed have given me a greater understanding of the software development lifecycle, as well as the amount of integration required to get the SLS Program off the ground. As my work continues I will be able to learn more critical information about the display development, meeting milestones and deliverables, subsystems and remote application software.

Acknowledgments

I would like to thank Kurt Leucht for directing me through the necessary training for the internship. Thanks to Cheryle Mako, Greg Clements and Heri Soto for being hands-off, this allowed me to take more initiative with my work. Additional thanks to Marcus Baldini and Judd Bishop for giving me task to complete that were important and great learning experiences. Lastly, thanks to my fellow Display Developer interns, Gentle Calhoun, Kristy Quaranto and Abner Coimbre for overall support and direction. Special thanks to Camiren Stewart for acting as "big brother" by showing me various nuisances of navigating the professional environment.

References

[1] LO2 SRDS (K0000064743-GEN), KSC Internal Document
[2] LCS ILOA Overview, KSC Internal Document
[3] ILOA Display Editor and Display Test Driver Overview, KSC Internal Document