

# Command, Control and Communications capabilities enabling 21st Century Missions, a historical perspective

Robert D. Waterman<sup>1</sup> and Herbert D. Rice<sup>2</sup>

*Architecture Integration and Management, Kennedy Space Center, FL, 32899*

Susan J. Waterman<sup>3</sup>

*IT Project Management Office, Kennedy Space Center, FL, 32899*

Command, Control and Communications (CCC) has evolved through the years from simple switches, dials, analogue hardwire networks and lights to a modern computer based digital network. However there are two closely coupled pillars upon which a CCC system is built. The first, is that technology drives the pace of advancement. The second is that a culture that fosters resistance to change can limit technological advancements in the CCC system. While technology has advanced at a tremendous rate throughout the years, the change in culture has moved slowly. This paper will attempt to show through a historical perspective where specific design decisions for early CCC systems have erroneously evolved into general requirements being imposed on later systems. Finally this paper will provide a glimpse into the future directions envisioned for CCC capabilities that will enable 21st century missions.

## A. Introduction

As we step through the changes in our nation's space program, from the first steps initiated by the Constellation program with its daring goal to return humans to the moon and then propel them onto Mars; to the bold new steps of NASA's changing direction, we should reflect on where our journey has taken us. From test flights launched from mosquito infested swamps, to man's first footsteps on the moon. From the transition of the Apollo program to the Space Shuttle program, through Shuttle sustainment and upgrades, to the forward looking strategies of the Exploration program. We are now perched at the brink of change. We must be prepared to execute our nation's new vision for space exploration and we must be equipped to adapt quickly and seamlessly to whatever direction it may take us. What capabilities will be required as we move from a large scale human launch capability to one of commercial growth and expansion? What will be required for heavy lift capabilities? What will be the role of a government space agency? Are there any foreseeable speed bumps in our path? Will this path be the one we sustain or are we destined to alter course with ever changing political goals? Or, will this be the trigger that catapults mankind forever into space, to travel farther and farther from our home world? Will history judge this new path to be the defining moment in space exploration where mankind made the tough choices that will inevitably preserve our species? As we prepare to walk this new path, the only thing we know for certain, is that we do not know anything for certain. As we move forward we must continue to push the boundaries of technology. However, NASA must also advance its behavior, its norms, and its culture. The NASA of today cannot succeed without becoming the NASA of tomorrow.

---

<sup>1</sup> Command, Control Architecture Lead, Architecture Integration and Management, LX-S3, non AIAA Member

<sup>2</sup> Communications Architecture Lead, Architecture Integration and Management, LX-S3, non AIAA Member

<sup>3</sup> IT Systems Project Manager, IT Project Management Office, IT-F, non AIAA Member

## **B. Historical Perspective**

The development, sustainment, and operations of Command, Control, and Communications (CCC) systems at the Kennedy Space Center have been essential since the arrival of the very first rockets; when teams of engineers and technicians carried the CCC hardware in the trunks of their cars from the Redstone Arsenal in Huntsville, Alabama to support an emergent launch campaign in Florida. As rocket complexity grew and the number of launches increased, reinstallation and revalidation of this mobile CCC system became an increasing schedule risk. It soon became apparent that a local, resident CCC capability was required. In the early days of this in-house CCC system, control actions were manual in nature. Human launch controllers managed the launch process with a hardware-only system of switches, gauges, lights and dials that required a dedicated human interface to perform every function, throw every switch, and verify every light and gauge. This approach was an acceptable solution due to the relatively simple systems that were being controlled and to the technologies that were available at the time. During the Apollo program, simple automation through hardware relay logic was added. Digital logic was incorporated during the Shuttle program, providing the means for control to be performed by software. Advancements in software capability would, over time, drive the need for increased hardware performance. Ultimately this need could not be met by the aging CCC system, causing the software capability to plateau because of constraints imposed by the existing hardware performance characteristics. In turn, when old hardware was replaced by new more advanced technology, it was constrained so that it did not impact the existing software pedigree. This set off a vicious cycle where changes to either hardware or software were constrained by either the existing hardware characteristics or the known software pedigree. The fallacy of changing either the software or the hardware but not both, when hardware and software is tightly coupled, is that the net system capability remains unchanged.

### **A. Apollo**

As human spaceflight drove the development of larger and more complex launch vehicles, the number of actions that needed to be performed for launch increased, while the timing between those actions became more critical thus forcing the development of automated ground processes for the Saturn rocket. In the era of Apollo, computational devices were rare; and while some did exist, they were not utilized to automate critical ground control functions. This type of automation was accomplished through hardware relay logic that was straightforward enough for the engineers to operate and troubleshoot; however, it did present challenges in testing the numerous electrical connections from the Apollo vehicle to the ground systems.

### **B. Space Shuttle**

During the time period between the last Apollo mission and the beginning of the Space Shuttle program, technology advanced to the point where mini-computer based systems were more common. However a commercial CCC capability that could be used to perform hazardous man-rated launch operations did not yet exist. This led to the development of Kennedy Space Center's Launch Processing System (LPS).

### **C. Launch Processing System (LPS)**

Kennedy Space Center was awarded the task of developing a CCC system to support ground and launch processing of the new Space Shuttle. Unlike previous launch vehicles, the Space Shuttle was designed with digital computers that interfaced to the ground launch processing system via a digital launch data bus. Designing the launch data bus protocol and its digital interface to the onboard computers was a challenging task, especially since the onboard system was developed concurrently with the ground launch processing system. Small changes to the onboard system could cause large impacts to the ground design. In addition, designers were faced with the challenges of developing a mechanism for interfacing digital ground system components and an approach to passing commands and data through the system in a deterministic manner. The CCC system that was eventually deployed was referred to as the Launch Processing System (LPS) and consisted of three major components. The first was the Checkout, Control and Monitoring System (CCMS), which was the heart of the command and control system. The second was the Central Data System (CDS), which supported data recording and retrieval of processed data. The third was the Record and Playback System (RPS), which recorded all raw data. For CCMS, each console in the firing room and every frontend interface to either Ground Support Equipment (GSE) or the Space Shuttle was connected through a point-to-point system which used a common data buffer as its prime communications channel. The advantage of this system was speed and deterministic behavior that ensured every message followed a known

and repeatable path. Using a common data buffer as the communication mechanism was an innovation for 1970's technology; however, today this approach severely limits the scalability of the LPS.

Prior to delivery of the LPS, several single point failures were identified. One of the most critical was a failure of the common data buffer, which could result in a loss of ground control due to the inability of the CCC system to communicate with the Space Shuttle and GSE. The solution that was implemented to mitigate this risk was to add a hardwired emergency safing system, which bypassed the digital CCC system and provided limited emergency control of Shuttle and GSE end-items. Though in hindsight, a major failure of the CCC system requiring the use of hardwired safing has never occurred; however, time and effort continues to be expended testing it prior to each launch. In the years since the initial Space Shuttle flight the actual hardwires have been replaced by a network-based system that does not utilize the primary command paths. Many LPS users have expressed a belief that hardwire safing is a salient part of any CCC system, and insist on its inclusion without analyzing the requirement to include it. As such, the hardwire or emergency safing system has found its way into the requirements of every CCC project under development since the creation of LPS, including those where other design alternatives existed.

Another deficiency of the LPS system is that it was designed with a custom software language that was developed in-house to allow operations engineers (operators) to directly specify command and response behavior without them needing specialized software development skills. This provided the benefit that operators could troubleshoot both the vehicle and GSE as well as the software that ran in the CCC system. Today this software seriously inhibits the ability to develop complex software algorithms for advanced processing such as control using frequency domain interaction instead of the traditional time domain, or analysis of system state by implementing advanced cluster or set theory.

With any system that has operated for more than thirty years, obsolescence has been and continues to be an issue with the Space Shuttle LPS. Through the years, hardware subsystems have been replaced in a piecewise manner with new technology that has been modified to operate exactly like the original equipment. This means that in most cases the capabilities inherent in the new technology have been intentionally hobbled and reduced in speed so that the rest of the system will not be impacted. The net result is that architecturally the LPS has not changed since its inception in the late 1970s.

#### **D. Checkout and Launch Control System (CLCS)**

In the late 1990's, an attempt was made to replace the aging LPS with a new Checkout and Launch Control System (CLCS). One of the goals of this system was to solve the obsolescence issues facing LPS by replacing aging less efficient hardware, and utilizing modern software development techniques. Another goal was to concurrently design and develop hardware and software throughout the core infrastructure; eliminating the problems associated with the piecewise changes made to the LPS system.

While no CCC system is perfect, CLCS suffered some of the same issues many development efforts experience when the goal is to replace an operational system. First, the existing LPS was mature with more than 30 years of runtime; and its operators were comfortable with its problems because they were largely considered to be known, understood, and characterized. Secondly, CLCS represented wholesale changes to the CCC architecture and would have required the application software to be rewritten using a language that was not easily understood by most operators. And finally, the operators were uncomfortable with such sweeping changes; partly because the "black box" technology made them feel uncertain that the requirements LPS fulfilled could be met and partly because understanding the code required specialized software development skills.

Refactoring an existing critical control capability poses two interesting dilemmas. First, the users/operators often insisted on what was familiar, that the new system work exactly like the old one; however, this negated most of the technological advancements that could have been implemented. Unfortunately, the typical compromise was to design the system such that it provided CCC capability that was as close as possible to what the current system offered with a caveat that new capability would be added at some later date. This approach only served to continue the culture of piecewise CCC design and development, thus fostering the same problems associated with technology gaps between hardware and software implementations. The second dilemma is that claiming a new CCC architecture improved safety was often viewed by others as substantiating that the old system was not safe. This statement could not have been any further from the truth than saying a car without anti-lock brakes should not be driven. When using

the old system, conditions that pose high safety risks are mitigated through alternative means. For instance, a person who wants to drive a car on slick snowy roads would choose the car with the anti-lock brakes; however, on dry nominal driving surfaces it is a wash, both cars are equally safe. Similarly, updating a CCC system to operate with up-to-date software and hardware technologies provides the ability to safely launch under certain conditions that were not acceptable with the old system. For example, because a critical common data buffer failure could cripple the ability of the LPS to communicate with the Shuttle and GSE, a hardware safing system was mandated. Suppose a new CCC system was developed such that this type of single-point failure was covered by redundant path software safing? Does hardware safing really need to be included? These two dilemmas stifle the development of upgraded CCC capabilities because the question that is inevitably asked if the system cannot claim it improves safety and is designed to emulate the old is "Is this endeavor worth the time, effort, and money?"

As the Space Shuttle program winds down, it will end using the same LPS architecture that was developed for its maiden flight more than thirty years ago. If the next program has comparable stamina and longevity, the same issues with obsolescence, implementing new technology, and applying innovative software solutions, will eventually resurface if not addressed upfront.

### **C. Current Activity**

#### **A. Constellation**

When the national space policy shifted to the vision for space exploration, NASA was presented with a unique opportunity. For the first time in thirty years, NASA would not be replacing or upgrading an existing CCC system, it would be creating a new one from the ground up. By definition, change was required, NASA would no longer be flying Space Shuttles; it would be launching new classes of rockets and spacecraft. NASA began this endeavor by defining the "Needs, Goals and Objectives" of the Constellation program and then by developing a set of architectural requirements. Lessons learned from the previous fifty years of launch history were reviewed and a top down architecture that included Command, Control, Communications, and Information (C3I)<sup>i</sup> was developed. For the first time, the CCC architecture considered an Information component to address data design, metadata (data attributes) design, and relationships between the ground and flight elements.

#### **B. Launch Control System (LCS)**

The Launch Control System design effort was based on three major architectural tenets. First, the system had to be sustainable for forty years. Second, the system had to be standards-based and not tied to a single vendor. Third, the system had to be designed for flexibility and adaptability to support the requirements of future spacecraft and launch vehicles that could not be articulated in the present. To achieve this, several trade studies were performed and a proof-of-concept<sup>ii</sup> system was implemented to ensure that the architectural tenets had a viable implementation approach. Ultimately the LCS architecture was designed around the Data Distribution Service (DDS) standards approved by the Object Management Group (OMG). LCS represented a major shift in strategy for designing CCC systems at the Kennedy Space Center. This was the first project to fully embrace Commercial-Off-The-Shelf (COTS) software in a critical CCC system. One of the proof-of-concept goals was to develop at least two implementations of each major component and then validate that the architectural tenets could be implemented using Commercial-Off-The-Shelf (COTS) products. A large majority of the development effort revolved around integrating various COTS capabilities into a unified system, while ensuring that the design did not rely on any specific vendor. While the individual LCS components were implemented with COTS products, the architecture itself was COTS product neutral; meaning that the system characteristics were defined without consideration of specific vendor capabilities.

The Standards Based Architecture (SBA) shown to be viable through the proof-of-concept activity was then baselined and the LCS project began full development activities. During development, each Computing System Configuration Items (CSCI) team performed trade studies, if necessary, to evaluate different solutions that could satisfy the system implementation; evaluating not only COTS solutions but also Government-Off-The-Shelf (GOTS), Modified-Off-The-Shelf (MOTS) and Open Source Capabilities. In most cases, this evaluation resulted in the selection of COTS capabilities to fill those needs. Even though many CSCI teams opted to utilize COTS

solutions, open source will play a significant role in the future. Open Source software is the middle ground between the cost of large in-house development efforts and the cost of sustaining and licensing COTS solutions. The trade-off when using COTS solutions is that they can offer lower development costs but they often require large licensing costs to sustain. Open Source on the other hand, often requires higher development costs but provides containment of sustaining costs. In the future, the manner in which open source software is managed, tested and certified may tip the scales away from COTS.

Finally, LCS embraced the Information Architecture<sup>iii</sup> proposed by the Constellation program and the concept of modeling the information and the relationships between the information in an ontology model. This model was then used to derive all of the build products that the LCS system needed to properly configure each of the CSCIs. The benefit of this approach over previous CCC system implementations was that changes to the information could be managed much more efficiently. Data updates to the Information Architecture would be converted to data files through an automated build process. In this way, the LCS system could adapt and change without requiring source code to be updated and re-verified.

#### **D. Future Directions**

LCS was designed and built as a generic CCC infrastructure. In fact throughout its development, the following principle was used to determine the system capabilities: "LCS should be able to launch a rocket or make toast," depending on the application software that is loaded on it. The LCS system was not only built to be used by Orion and Ares but also to be adapted to other spacecraft. As far as LCS was concerned it was communicating with a network appliance that spoke a dialect of C3I. The three tenets of LCS have not changed. It was designed upfront for longevity. It was designed to be sustainable beyond the lifetime of any individual vendor or technology. And it is able to support future customer needs that cannot be articulated in the present.

Looking into the future, at least three roles for NASA can be postulated where specific capabilities of a CCC system would be needed. The first role is one of NASA insight and mission assurance. If commercial entities launch from a NASA-owned launch pad, some level of resource protection would drive a minimal monitor-only capability. The second role would be one of supporting test flights for a Heavy Lift Launch Vehicle (HLLV). For the HLLV it is assumed that NASA would need the ability to command the launch vehicle. The third role is one of providing support to commercial launch services. While the exact services required to provide a commercial launch provider the best opportunity for success are currently unknown, educated guesses can be made until an opportunity exists to make direct inquiries. Some of the capabilities that would be needed are the core infrastructure inherent in launching from a NASA launch pad, such as, voice communications, imagery, paging and area warning, network infrastructure and basic command and control.

The following questions represent example capabilities being discussed to further bound the CCC space. Will a commercial launch entity want to bring their launch team to KSC and launch out of the Launch Control Center (LCC)? Should KSC support virtual launch teams? If a commercial entity has an existing CCC system, should NASA provide the capability to remotely connect to the launch pad from anywhere on the planet; why bring the launch team and the CCC system to KSC? Can NASA provide any additional services to augment the commercial customer's existing capability? Should NASA provide other capabilities such as: manufacturing support, post liftoff capabilities, or launch expertise? What unique capabilities should NASA provide to ensure that each commercial entity is successful? These higher level capabilities, while postulated, are awaiting future discussions to determine implementation and prioritization strategies.

#### **E. Conclusion**

As discussed in this paper, NASA has amassed a large experience-base for performing critical Command and Control of launch vehicles and ground systems. And as NASA continues move forward, it is important to assess past CCC systems for Lessons Learned; capturing both successes and failures. Requirements should be reevaluated for applicability and retained if relevant and thrown away if not. For example, if LPS developed a hardwire safing system to solve a single point failure of the common data buffer, that does not imply hardwire safing should be an implicit requirement of future CCC systems. Requirements for new or upgraded systems should not insist that requirements from past systems be incorporated unless an independent evaluation deems them necessary.

The state of CCC system design has evolved from manual controls, to hardware-only controls, to digital and microprocessor-based controls, to those of today where software performs a majority of the system functions. Because of this, it is no longer sufficient to design CCC hardware in a vacuum; the software must be architected and designed concurrently, especially if system components are expected to interoperate.

Architectural and cultural impediments to implementing advanced and/or complex software in future CCC systems must be tackled. The CCC hardware and software should be decoupled such that changes or upgrades made to one should not impact the other. Operations engineers should not have to understand the CCC software in order to operate the system. In other words, system operators should treat the software like a “black box” that is sustained by personnel with software development skills who are responsible for performing upgrades and troubleshooting anomalies.

To successfully support a commercial launch capability, it is imperative that NASA as an agency asks its customers, “What can we do to make you successful?” This single question must be the foremost consideration for charting a path to success. As NASA embraces this new commercial launch services direction, our future success will be judged not on how well we perform, but ultimately on how successful new commercial space initiatives are. It is time for a new group of commercial operators to be allowed to earn their wings. They must be successful, for NASA and this country to have any chance of succeeding at exploration of space beyond Low Earth Orbit. We must adapt and change our behavior if we wish to have a relevant role in the future. As Norman Peale, the author of *“The Power of Positive Thinking,”* once said: “If you want things to be different, perhaps the answer is to become different yourself.”

### **Acknowledgments**

The authors of this paper would like to thank the members of Kennedy Space Center’s Command, Control, and Communications sub-team for their help in formulating and promoting the Vision to Support 21<sup>st</sup> Century Missions. In addition, we would like to thank the many contractors and civil servants who have tirelessly worked to develop CCC systems at KSC throughout the years. To paraphrase Sir Isaac Newton, if we have a better vision for the future now, it is because we are standing on top of all the work others have done to successfully support NASA space launches.

### **References**

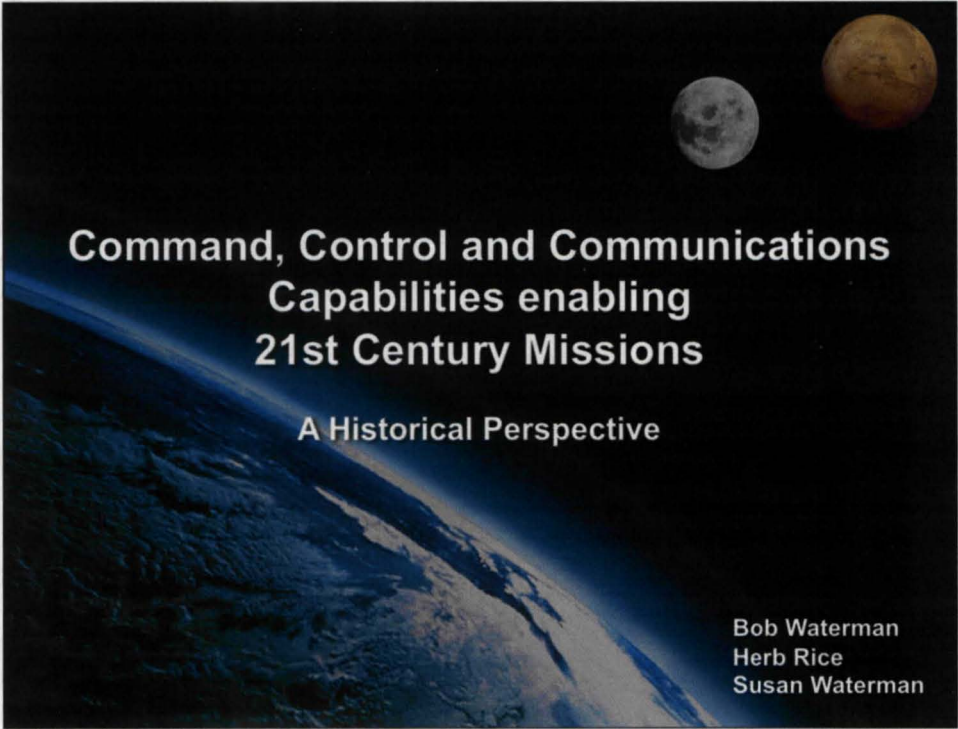
---

<sup>i</sup> Constellation program Command, Control, Communication, and Information (C3I) interoperability standards book, volume 1: interoperability specification CxP 70022-01, NASA Lyndon B. Johnson Space Center, December 14, 2009

<sup>ii</sup> Ground Elements Command, Control & Communications Project Proof-of-Concept Findings Report GOP-507023, NASA John F. Kennedy Space Center, June 29, 2007

<sup>iii</sup> Constellation program Command, Control, Communication, and Information (C3I) interoperability standards book, volume 4: information representation specification CxP 70022-04, NASA Lyndon B. Johnson Space Center, August 12, 2009





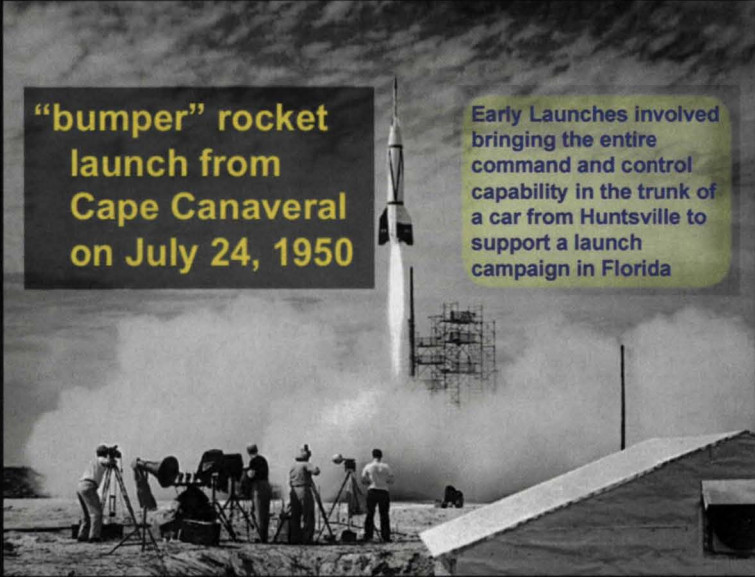
# Command, Control and Communications Capabilities enabling 21st Century Missions

A Historical Perspective

Bob Waterman  
Herb Rice  
Susan Waterman

1950s

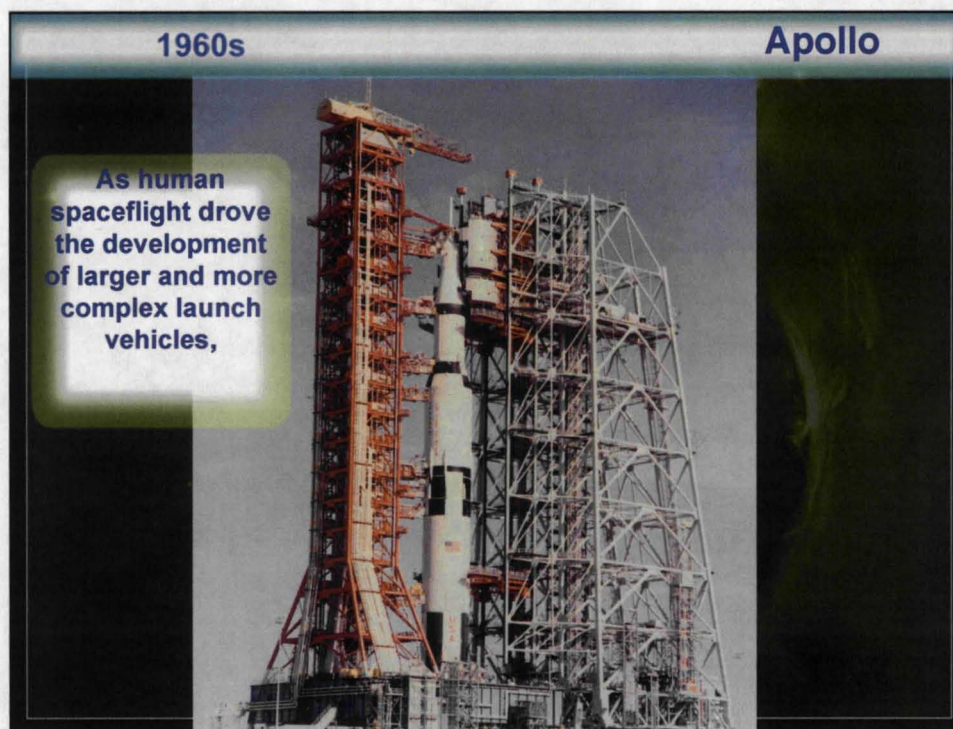
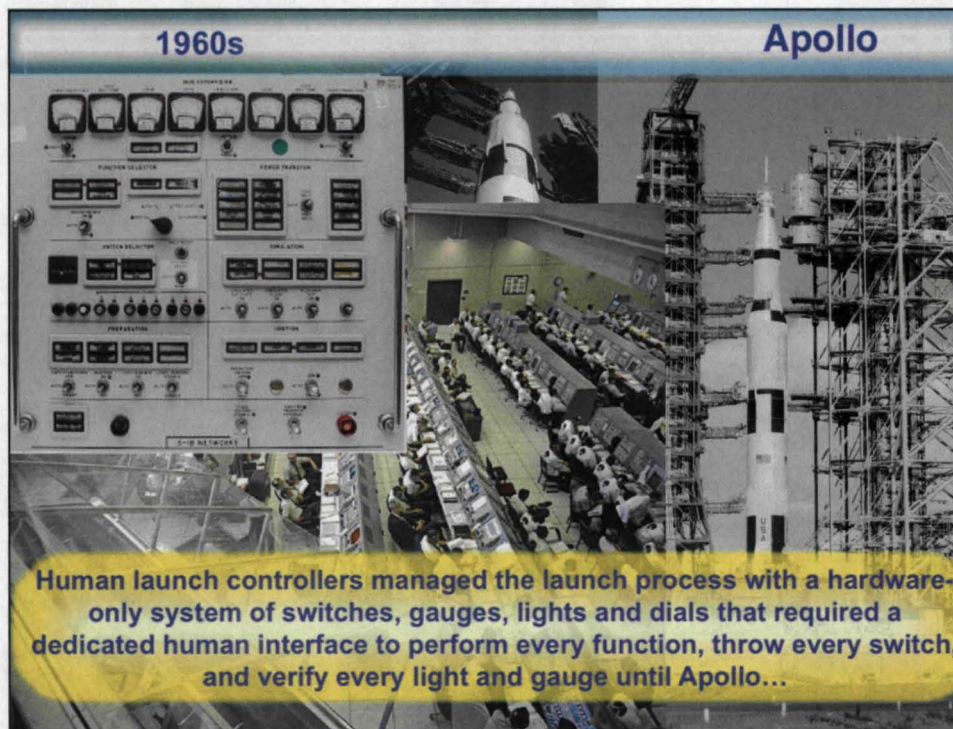
## The Beginning



**“bumper” rocket  
launch from  
Cape Canaveral  
on July 24, 1950**

Early Launches involved  
bringing the entire  
command and control  
capability in the trunk of  
a car from Huntsville to  
support a launch  
campaign in Florida



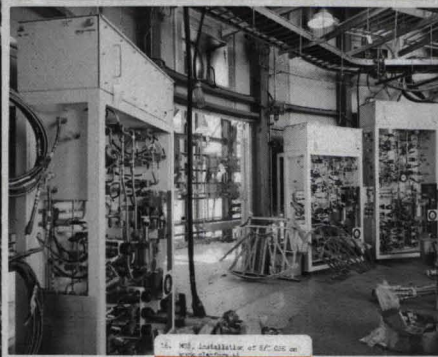
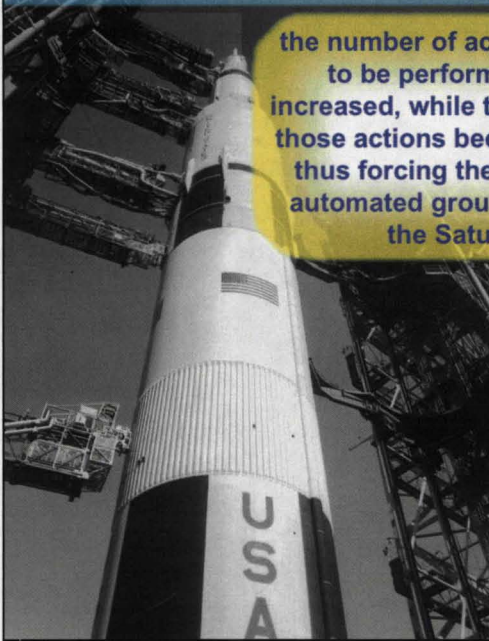




1960s

Apollo

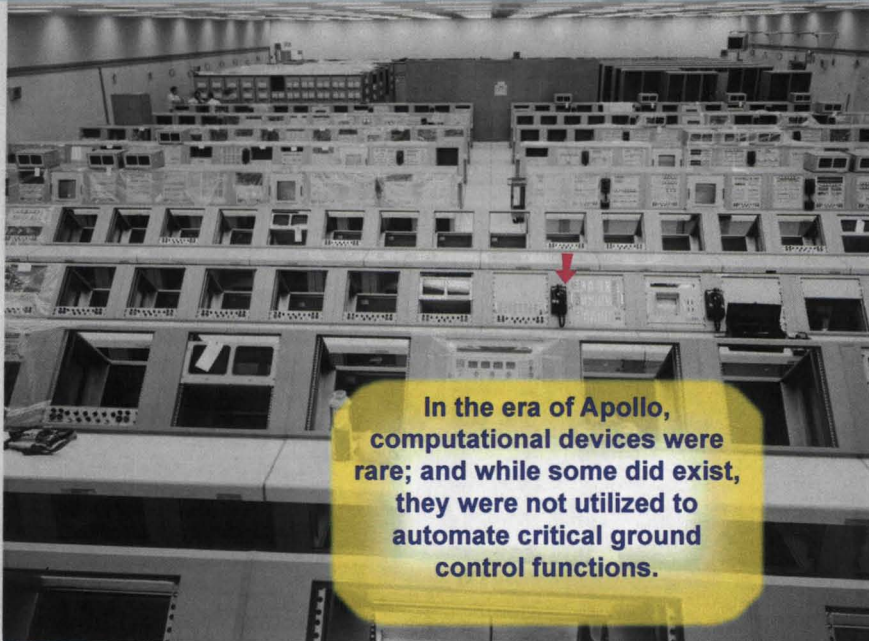
the number of actions that needed to be performed for launch increased, while the timing between those actions became more critical thus forcing the development of automated ground processes for the Saturn rocket



1960s

Apollo

In the era of Apollo, computational devices were rare; and while some did exist, they were not utilized to automate critical ground control functions.





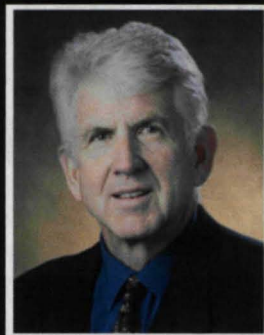
1960s

Apollo

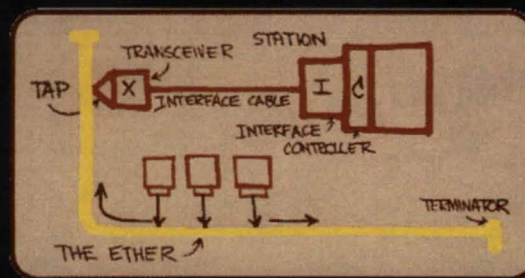


1970s

Space Shuttle



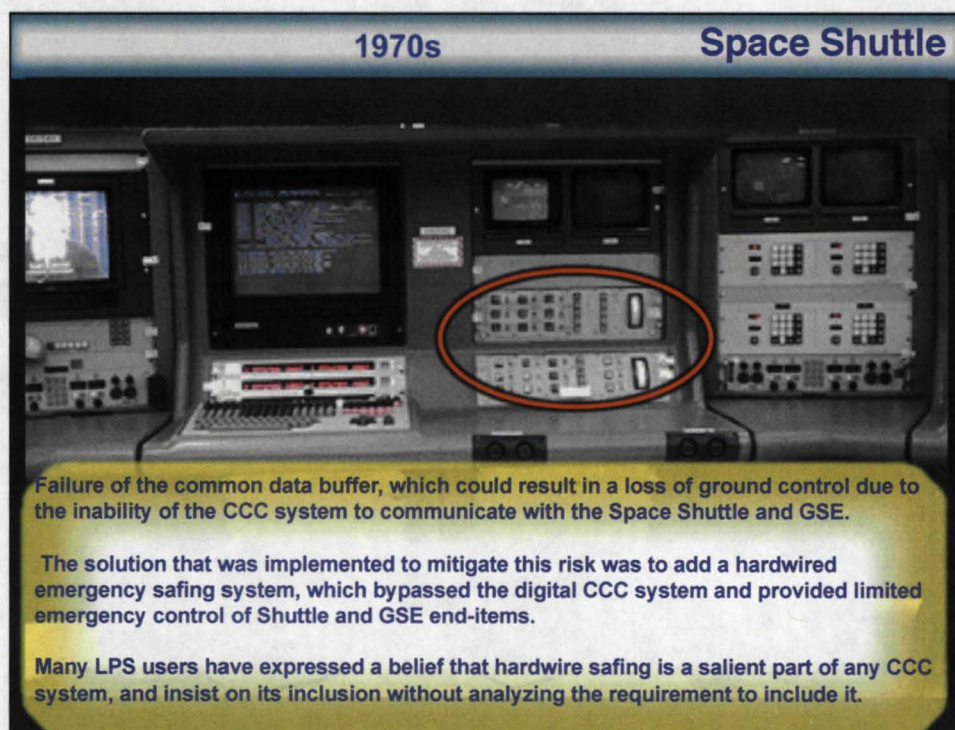
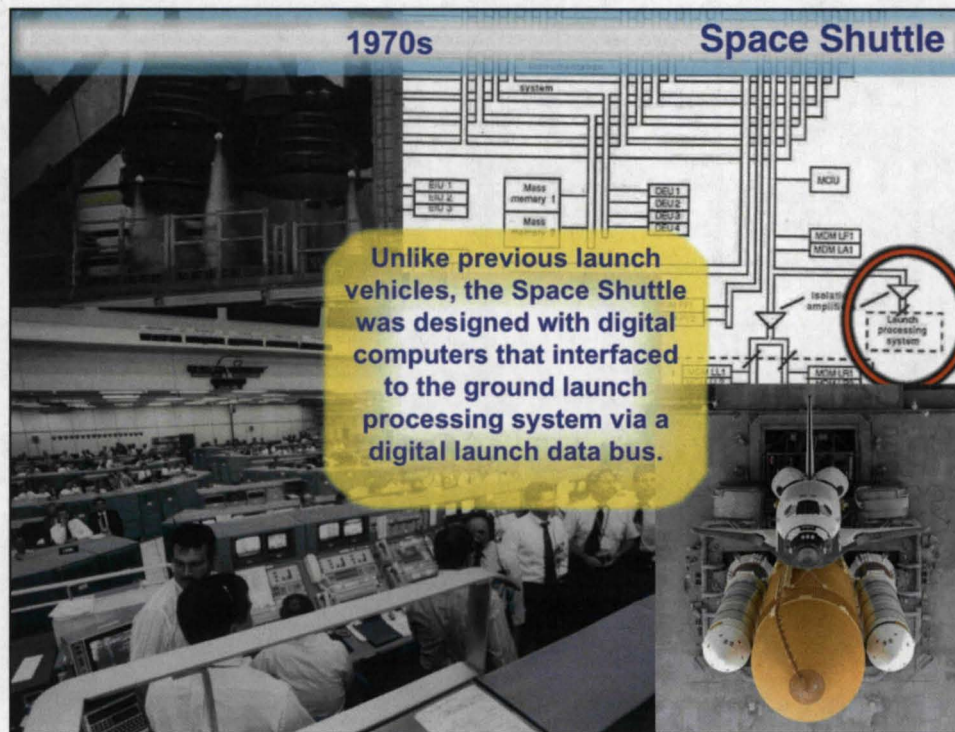
\*



About the same time that this distinguished individual created the concept sketched above. NASA had to design And deploy a human rated, network based Launch Control System

\* Robert Metcalfe - Co-Inventor of Ethernet and later founder of 3COM





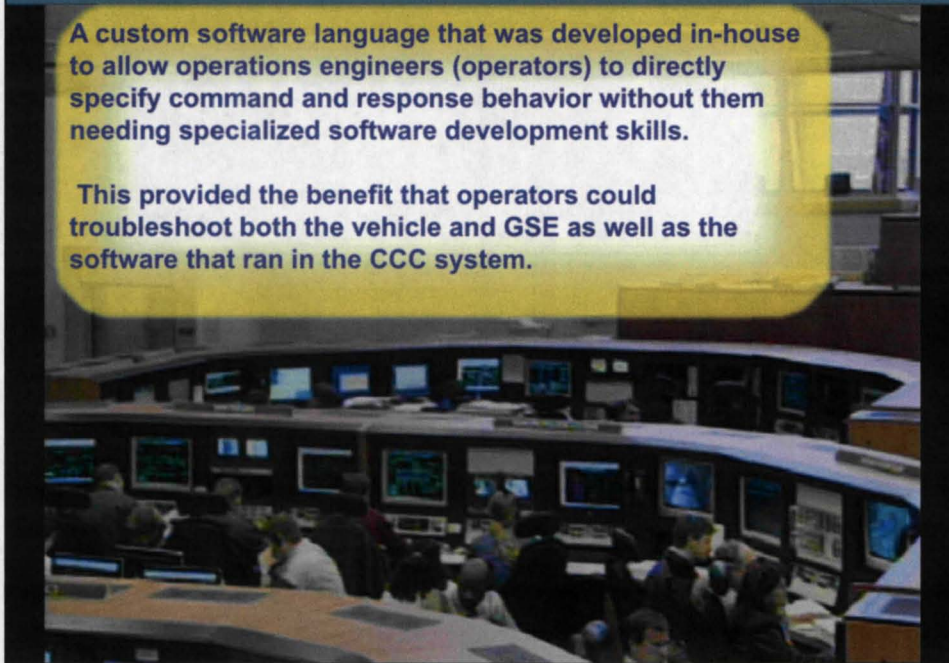


1970s

Space Shuttle

A custom software language that was developed in-house to allow operations engineers (operators) to directly specify command and response behavior without them needing specialized software development skills.

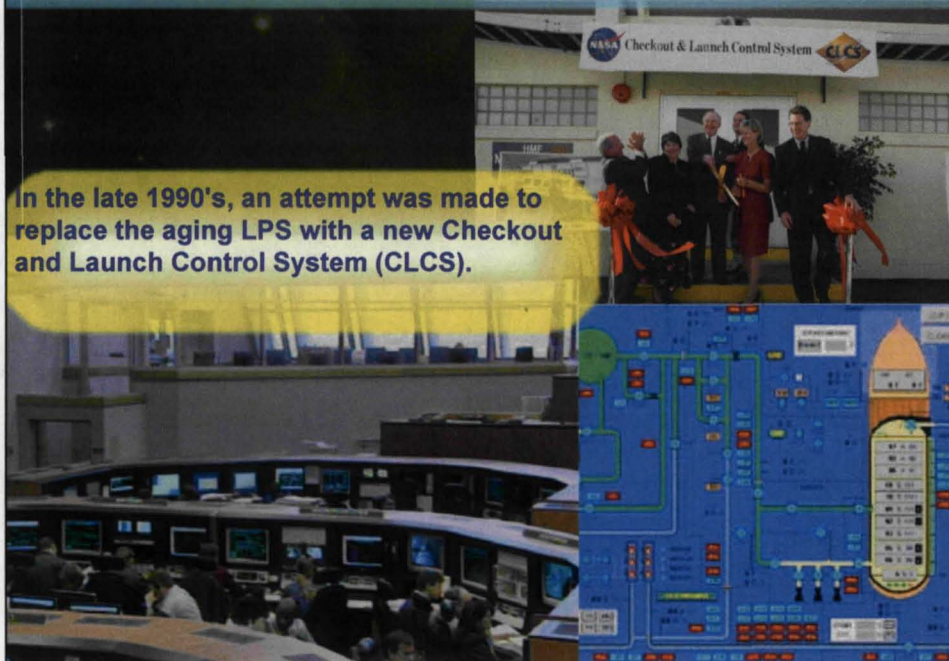
This provided the benefit that operators could troubleshoot both the vehicle and GSE as well as the software that ran in the CCC system.



Space Shuttle Upgrades

1990s

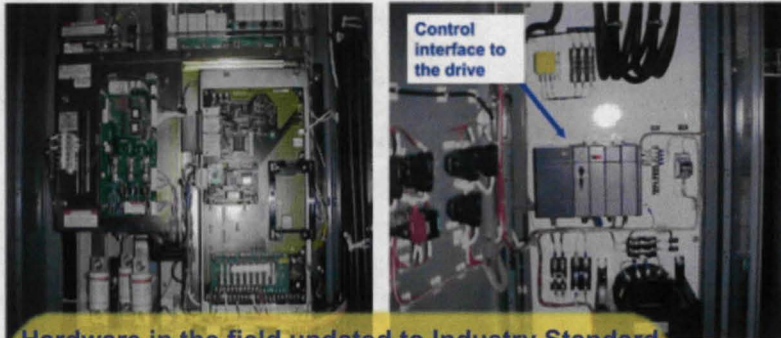
In the late 1990's, an attempt was made to replace the aging LPS with a new Checkout and Launch Control System (CLCS).





### The field hardware example

- Liquid Oxygen ( LOX) Variable Frequency 300 HP Pulse width Modulated Drive

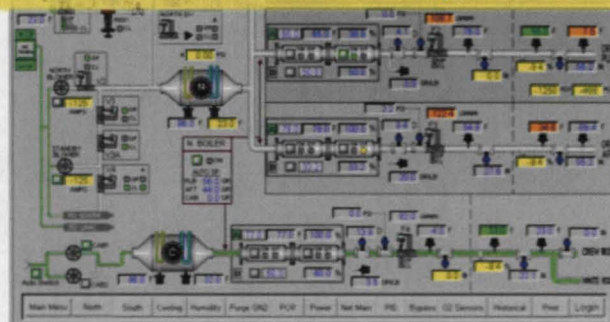


Hardware in the field updated to Industry Standard Based Control technology

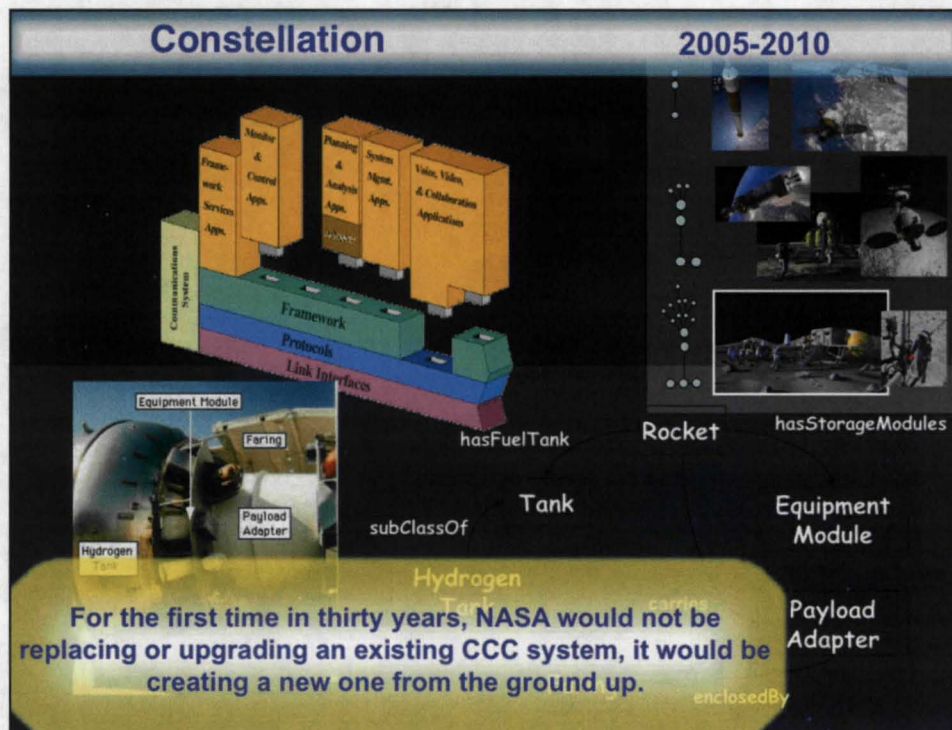


### The "Front end" Supervisory Control for ECS and Health Management Network

Industry standard  
Supervisory Control and Data Acquisition system  
Implemented using  
Commercial off The Shelf systems







## Constellation

2005-2010

The Launch Control System design effort was based on three major architectural tenets.

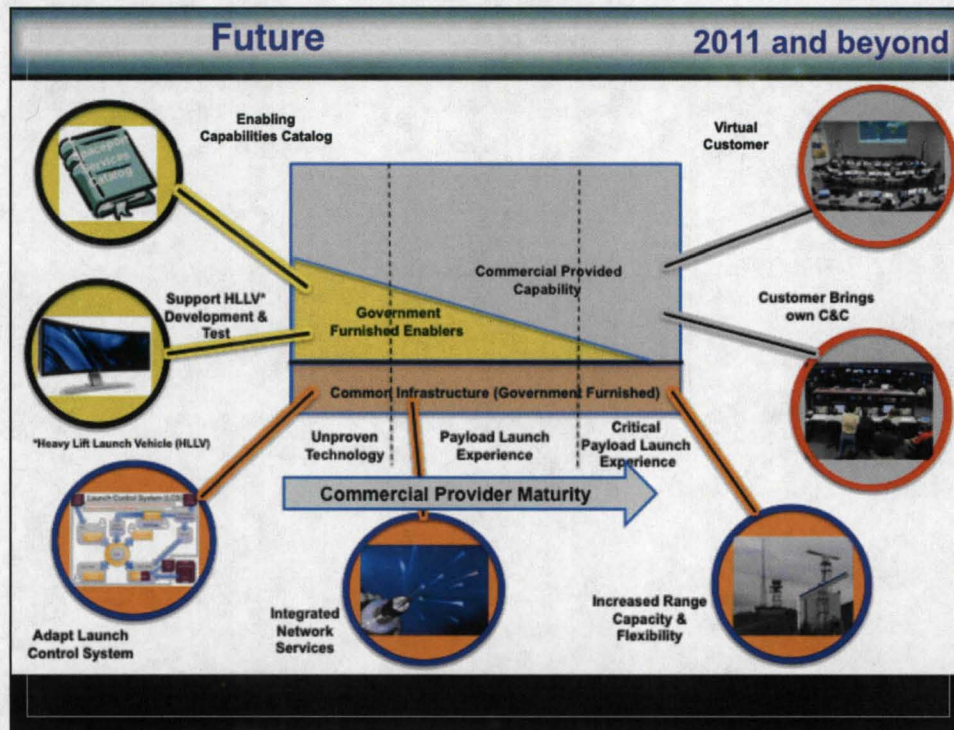
First, the system had to be sustainable for forty years.

Second, the system had to be standards-based and not tied to a single vendor.

Third, the system had to be designed for flexibility and adaptability to support the requirements of future spacecraft and launch vehicles that could not be articulated in the present.

**Launch Control System**





**Future** **2011 and beyond**

## Welcome to the Florida Spaceport

Today is Tuesday April 14, 2020

Commercial Crew launch from Pad 39-B to ISS at 8:47 AM	<b>ON TIME</b>
Mars Communications Satellite launch from Pad 41 at 1:13 PM	<b>ON TIME</b>
Commercial satellite launch from Pad 37 at 6:35 PM	<b>ON TIME</b>

**Tomorrow April 15, 2020**

Space Hotel passenger launch from Pad 39-A at 11:00 AM	<b>ON TIME</b>
--	----------------

*Notice: Use of Virtual Reality sensory implants while traveling on Florida's Spaceport is illegal*



**"If you want things to be different,  
perhaps the answer is to become different yourself."**

*Norman Peale,  
Author of "The Power of Positive Thinking,"*

