Monday, January 09, 2012

Table of Contents

Tuesday, October 17, 2017

GEMINI PROGRAMMING DEVELOPMENT AND VERIFICATION PROCESS

1.0 INTRODUCTION

This document was written to identify the process of Developing and Verifying Gemini Computer Programs (called Math Flows- MF) in 1962 through 1966. It was written by Pat Mooney, the Gemini Programming Manager and Charlie Leist, (Re-Entry and Touchdown Predict Programmer) in January 2012, 45 years after the last Gemini GT-12 Flight.

The Gemini Peer Review Process of Reviewing and Inspecting Gemini System Flow Diagrams, Detail Design Flow Charts and Program Code just before Math Flow ship to MAC was a key step in delivering error free programs. In 1976, Michael E. Fagan, from the IBM Commercial Division, wrote a paper in 1976 on “Design and Code Inspections to reduce errors in Program Development” using a similar approach utilized on the Gemini Project.

2.0 GEMINI PROJECT

The Gemini Project started in 1962 and ended in 1966. Twelve successful missions were flown. The on-board Gemini computer had operational programs to perform mission objectives:

1. Ascent Guidance, from lift off to entering orbit
2. Catch – Up and Rendezvous, mating up with another space vehicle
3. Re-entry, firing Retro rockets to de-orbit and provide guidance for returning to earth

The computer operational programs were called Math Flows (MF). See Table 1 for the relationship of MF to Gemini Missions. The Gemini MF on board code was generated and supported by programmers and system engineers shown in Appendix A.

Programming the Gemini was a new science and had to be created, learned and procedures/standards generated.

The IBM Owego people working to develop the Gemini Computer and computer programs were driven to provide excellence in their work products for the following reasons:

3.0 NATIONAL GOALS

President Kennedy-Land a Man on the Moon by the end of 1979

NASA Administer James Webb - Keep Astronauts out of harm's way that could affect their lives

4.0 IBM's PRINCIPLES

Respect for the Individual

Service to the Customer

Excellence Must Be a Way of Life

With these Goals and Principles always in the mind of the IBM Gemini Programmers, it made their design and products meet high Quality Standards and generate error free code to support mission requirements.

The programmers accomplished this task by generating Detail Math Flows from the System Math Flows (program requirements) provided by the System Engineers and developing Gemini Program Code from the Detail Math Flows.

A rigorous static and dynamic test activity followed program coding to identify errors and remove them for retest.

To eliminate any remaining errors a peer review/inspection process was developed and implemented just before shipping the Gemini programs for the next level of integration with the Gemini Spacecraft at McDonnell-Douglas (MAC) in St. Louis and at the Cape on the pad. The process included Planning, Preparation, Peer Review Meeting, Rework and Re-Review meeting/Error Sign off.

5.0 GEMINI MATH FLOW AND MISSION TABLE 1

MATH FLOW	RELEASE	SELL OFF	MVS COMPLETE	COMMENT and MISSON
MF-1	11/62	5/63	None	Sold off in CCTS Lab
MF-2	6/63	None	None	Released to MAC as Preliminary
MF-3	11/63	5/15/64	1/65	Flown on GT-2
MF-3 MOD I	7/64	12/64	3/65	Flown on GT-3
MF-3Mod II	9/64	2/15/65	6/65	Flown on GT-4
MF-4	11/63	None		Work Stopped 6/64
MF-5	None			Work Stopped 6/64
MF-6	10/64	None	8/65	Flown on GT-5, GT-6, GT-7

MF-7	9-65	1/21/66	3/66	Flown GT-8, GT-9, GT-10, GT-11, GT-12

Tracing the development of the math flows shows how the Gemini Project handled changes.

1. Math Flow One consisted of just four modules: Ascent, Catch-up, Rendezvous and Re-entry
2. Math Flow Two was proposed to add orbit navigation and Re-entry initialization, but it caused the overall load to exceed the memory size and the Gemini Program Office canceled the additions. Math Flow Two flew on Spacecraft II in January 1965.
3. By Math Flow Four, the Re-entry initialization program had been successfully added, but the memory load took up 12,150 of the 12,288 available memory words. The plan had been to use this program on Spacecraft III and others, but a NASA directive of February, 1964 changed the guidance logic of the Re-entry mode to a constant bank angle rather than proportional bank angle and constant roll rate.
4. Math Flow Five incorporated the constant bank angle change, but it filled the memory and was scrubbed in favor of a modified Math Flow Three on Spacecraft III and IV.
5. Math Flow Six contained some changes on Spacecraft V through VII.
6. Math Flow Seven was used on Spacecraft VIII through XII all of which incorporated the Auxiliary Tape Memory. It had six program modules with nine operational modes of operation: Executor, Prelaunch, Ascent, Catch-UP, Rendezvous and Re-entry.

6.0 GEMINI COMPUTER

The Gemini Digital Computer is a binary, fixed-point, stored-program, general-purpose computer, used to guide the spacecraft. The computer is 18.90 inches high, 14.50 inches wide and 12.75 inches deep. It weighs 58.98 pounds.

Using inputs from other spacecraft systems along with a stored program, the computer performs the computations necessary to develop the guidance and control outputs required by the spacecraft during the Pre-Launch, Catch-up and Rendezvous, and Re-Entry phases of the mission. In addition, the computer provides back-up guidance for the launch vehicle during Ascent.

6.1 INPUTS and OUTPUTS

The computer is interfaced with the Inertial Platform, Platform Electronics, Inertial Guidance System (IGS) Power Supply, Auxiliary Computer Power Unit (ACPU), Manual Data Insertion Unit (MDIU), Time Reference System (TRS), Digital Command System (DCS), Attitude Display, Attitude Control and Maneuver Electronics (ACME), Titan Autopilot, Pilots’ Control and Display Panel (PCDP), Incremental Velocity Indicator (IVI) and Instrumentation System.

6.2 OPERATIONAL PROGRAM

The Operational program consists of six basic routines: Executor, Pre-Launch, Ascent, Catch-up, Rendezvous and Re-Entry. Each routine is made up of several subroutines. Some of the subroutines are common to a routine while some are unique to a particular routine. Each subroutine consists of a series of program instructions which, when executed, cause specific computer circuits t operate. The initiation of a particular routine is controlled by the Computer Mode selector switch on the Pilots’ Control and Display Panel. Once a routine is initiated, the subroutines within the routine are executed automatically.

7.0 PROGRAMMER DESIGN AND CODING INSTRUCTIONS

The following Gemini programming instructions were generated for use in developing the Detail Math Flow Diagram and Program Code. During the Peer Review and Inspection meeting these instructions were reviewed for compliance.

1. Each column on the Detail Math Flow is to start with a letter representing the module mode, (R1 .1 stands for Re-entry mode) in column 1, block 1.
2. Each block in a column is to have a number.
3. Each equation type block is to show the equivalent units on bottom right corner.
4. The dependent variable scaling is to be shown in the upper right-hand corner of each detailed math flow diagram block.
5. Every constant used on the detailed math flow sheet is to be defined and all constants are to be placed in the Gemini Symbols and Constant Document.
6. Every Left-Hand Symbol (LHS) referenced by the operational program is to be shown on the detail math flow.
7. All subroutine calls are to be standardized.
8. All variables are to be subscripted properly in the detailed math flow equations.
9. All detail math flow symbols and program symbols are to be explicitly defined in the Gemini Symbols and Constant Document
10. KZERO is a constant with all zeros. Use this when it is necessary to have all 26 bits with all zeros.
11. Fast loops (I/O) must be inserted in the Gemini Code with a HOP instruction every 35 ms so that the I/O subroutine, which takes 15 ms, will meet the 50 ms Ladder decay requirement.
12. The DCS and DAS request remains up for only 75ms. So, if the fast I/O loop exceeds the value of 75ms a DAS or DCS request will not be honored.
13. System math flow and detail math flow must be in agreement and traceable.
14. The detailed math flows must agree with the corresponding program listing.

15 Each equation in the detailed math flow is to be checked for proper scaling and proper unit assignments.
16 All constants are to be cross checked with the system flight constants and the program listing. Also, check program listing constants with the detailed math flow.
17 The system flight constant list is to be crossed referenced to the Gemini Symbols and Constant Document.
18 Check to see that all Variables are properly initialized or computed prior to their use in the Gemini operational program.
19 Program code comments are to be proceeded with a # sign.
20 Divide Instructions requires 5 instruction to get the answer. Double check.
21 Multiply instruction requires 3 instructions to get the answer. Double check.
22 HOP I/O must be placed in the code every 35 ms. Double heck.

Development of the Gemini programs was a learning experience for both NASA and IBM. It was, of course, the first on-board software for a manned spacecraft and was certainly a more sophisticated system than any that had flown on unmanned spacecraft to that point. When the time came to write programs for Gemini, programmers envisioned a single memory load containing all code for the flight with new unique programs to be developed for each mission. Such as the Ascent Guidance program that was used as a back up to the Titan launch vehicle.

8.0 SPECIFICATION

The specification requirements for the programs required McDonnell-Douglas to prepare the Specification Control Document (SCD). This was forwarded to the IBM Space Guidance Center in Owego, which developed a FORTRAN program to validate the guidance equations before they were coded in Gemini language.

9.0 SIMULATION

Gemini used three levels of simulations beginning with:

1. The equation-validation system
2. The man in the loop simulation to help define I/O requirements, procedures and displays
3. The third level was a refined digital simulation to determine the performance characteristics of the programs, useful in error analysis

This third level was carried out in the Configuration Control Test System (CCTS) laboratory, which contained a Gemini computer and crew interfaces.

The Mission Verification Simulation (MVS) ensured that the guidance system worked with the operational mission program in a closed loop fashion that provided realistic computer inputs from an environment program.

NASA and IBM emphasized program verification because there was no backup computer or backup programs. This verification process and tools developed for it were later applied to other military projects in which IBM became involved.

Even if programs are perfect, errors may occur because of transient hardware or program failures during operation due to power fluctuations or unforeseen demands on real-time programs. Some of these can be spotted by diagnostic subroutines interleaved in the program for fault detection. Such diagnostic routines were put in the Gemini executive routine and later became a part of all IBM computer systems.

10.GEMINI COMPUTER PROGRAM FINAL DESIGN AND PEER REVIEW PROCESS

Once a Gemini Operational Program had completed all development and testing activities a final peer design and code review was held on the Math Flow. This event required several days of work by the system engineers and programmers. The meeting was held in an IBM Owego conference room. The purpose of the peer inspection and code review was to detect any errors before shipment to MAC.

The Review team consisted of the responsible system engineers, programmers and a person to record findings. Other interested personnel were invited to attend and participate the meeting.

The Planning Phase included obtaining the review materials and inviting the attendees:

1. Specifications
2. System Math Flow
3. Detail Math Flow
4. Program Code listing
5. Symbols and Constants Document
6. Programming Manual
7. Engineering Notebook
8. Set up meeting and obtain conference room
9. Invite participants

The Peer Review and Inspection Meeting Phase

1. System engineer identifies the specification requirements and describes the system math flow compliance
2. These documents are inspected and reviewed for agreement
3. Programmer presents the Detail Math Flow and demonstrates that it is in compliance with the System Math Flow requirements and design on a block by block basis
4. Inspectors and reviewers perform their assessment of the detail design compatibility with the system math flow
5. Programmer presents the Assembly listing that shows the coded program and its compliance with the Detailed Math Flow

6. Inspectors and Reviewers evaluate on an instruction basis that logic paths, constants, variables, LHS, equations, I/O timing, initialization parameters, units and scaling are in agreement with the Detail Math Flow.
7. Errors detected are logged in the Engineering note book by the assigned data recorder person
8. Meeting is closed with action items assigned

The Error Rework Phase

1. The person assigned an identified defect makes a correction and documents the fix and retests if required to verify the problem is successfully solved

The Error Resolution Phase

1. A meeting is held to review action items for closure and review error closure data for accuracy
2. Engineering notebook is updated to show that errors have been fixed and signed off for closure
3. Peer Review and Inspection meeting is closed with recommendation to ship the MF is made

11. CONCLUSION

The programs produced during the Gemini Project were highly reliable and successful. Techniques of specification development, requirement verification, simulations and Peer Reviews developed for Gemini were later applied to other IBM and NASA projects. NASA was better prepared to monitor programming development for the Apollo Project.

Gemini Program Peer Reviews were the final step in delivering error free code and was the precursor of Design and code inspection process used by IBM Commercial Division in 1976 and documented by Michael E. Fagan.

The NASA and IBM verification process and tools developed for Gemini were later applied to other military projects in which IBM became involved.

The technique of embedding diagnostic routines in the Gemini executive control program later became a standard feature in all IBM developed computer systems.

APPENDIX A GEMINI PROGRAM PERSONNEL

People working on Gemini Project in Owego, NY (1962 to 1966) who interfaced with the programming team

FSD President

Bob O. Evans

General Manager

Art Cooper

Manager of Gemini Program Office

J. Clint Huntley

Other Managers

Clint Grace General Manager Huntsville

Hal Sohn Contracts

Jim Slavin Project Office

H.W. Hutchison Engineering

Conrad D. (Del) Babb Program Manager

Harry Branning System Engineering Manager

James Handley Titian computer programming manager

Ken Wadman Test Engineering

Dave R. Baldauf Management

Dick Hillsley Management Bethesda

Homer W Hutchison Management, Test Equipment

John Sweeney Management Computer design and hardware logic

Connie McClure Mathematician and Training

Programming Manager

Pat Mooney

Programmers

Jim Joachim – Group leader

Lee Jackson - Manual Data Insertion Unit (MDIU) and Diagnostics

Charles Leist - Re-entry and Touch Down Predict

Jim Condell - Hard Core, Executive and MDIU

John Elmes - Re-entry

Alden Minnick - Catch up and Rendezvous

Norm Kemerer - Simulation

Andy Gibson - MDIU

Sharon Koblinsky – Telemetry

Pat Tyron – Programming Support

Francis Scott Coffin - Simulator

Don O'nell- Assembler

Charlie Corvette – Diagnostics

Al Melgaard - Orbit Navigation

Dick Eckstrom – Orbit Navigation, Autonomous Navigation Programmer

Fred Kurz – Programming Support

Don Puro – Ascent Guidance

Carl Freitag – 7090/1401 Computer Room

Vinnie Zacheo – Computer Operator for 7090/1404

Joe Constable – FORTRAN Programmer

Len Masiowski – OBC Programmer

System Engineers

Roy Jimerson – Computer Engineering (wrote original proposal)

Dr. Robert J. Urquhart Creator of the original design of the computer\

Bill Patzer Engineering Gemini Design team member

Joseph J Shanis-Computer Design Engineer for Inertial Guidance

Gene Mertz-Catch up and Rendezvous

Dale Bachman-Re-entry

Tom McDonald-Re-entry

Jim MacPherson-Re-entry

Richard Jasinski-Ascent Guidance

Paul Beale – Ascent Guidance

A.J. Boardman – Touchdown Predict

Dick Higley- Specifications, Test Writer

Lee A Danielson- Inertial Guidance

Dal G. Ferneyhough, Jr.-Catch up and Rendezvous

Fred Reiner - Catch up and Rendezvous

Paul Obsharsky - Engineering

Marv C. Owen - Engineering

Henry Robson –Computer Engineer

Bill Morrison- Test engineering

W.D Coryell – Test Engineering

L.Liebschutz –Test- Engineering

R.C. Fisher – Test Engineering, Designed core memory unit with Bob Urghardt, CPU

Karl Blaine – IBM Rep at MAC and Cape

Will Robinson – IBM Rep at MAC and Cape

J.J. Foard – IBM rep at MAC

Lee A Danielson- Inertial Guidance

Walt Vogdes – Re-entry Guidance Simulator

John HG Burrell – OBC Hardware Expert

Homer Middleton – Module ‘N’ core map, OBC to ATM format

Robert F Miller – Test Specifications for Ascent

APPENDIX B REFERRENCES

1. GEMINI PROGRAMMING MANUAL   12-20-2011
2. Gemini Web Links
    a.