D45.

MANAGING COMPUTER - CONTROLLED OPERATIONS

John B. Plowden*
Shuttle Launch Operations Division
Rockwell International
Kennedy Space Center, Florida

ABSTRACT

A detailed discussion of Launch Processing System (LPS) Ground Software Production is presented to establish the interrelationships of Firing Room resource utilization, configuration control, System Build operations, and Shuttle Data Bank management.

The production of a Test Configuration Identifier (TCID), will be traced from requirement generation to program development, Data Bank update, GOAL program debug and verification to release thru Configuration Management for use in processing flight hardware.

The challenge of the Operational Era will be to implement fully automated utilities to interface with a CDS resident System Build Requirements Document to eliminate all manual intervention in the System Build operations. Automatic update/processing of Shuttle Data Tapes will enhance operations during multi-flow processing.

INTRODUCTION

Ground Software Production for processing and Launch of the Shuttle Vehicle begins with the generation of requirements for a specific mission. The changes that effect ground software can be a hardware modification, flight software modification, measurement change, or a checkout requirements change.

Requirements are processed through the Configuration Control Board where impacts are determined to effect OMI's, application software, etc. Based on approval, updates to the effected elements are scheduled. Automated tracking systems are maintained to provide visibility of open work items to support software planning, scheduling, and engineering open item reviews.

These automated tracking systems are the cornerstone of an automated Configuration Management (CM) system that efficiently tracks software requirements from identification through completion. The CM system deletes manual intervention by interfacing directly with the System Build process. The software selects the approved programs and console locations based on Vehicle (099, 102, 103), Site (OPF, VAB, PAD) and Facility (MLP1, MLP2, HB1). Additionally the CM system generates a computer maintained work authorizing document for Firing Room verification supporting real-time Quality Assurance buy-off, automated library updates and Release Notice generation.

The System Build operation utilizes the TCID in the compilation and translation of all application software elements into a controlled application library resident on the Central Data Subsystem (CDS). The following elements are included in a TCID: (1) Flight Format Files (defines valid Pulse Code Modulation formats), (2) Function Designator Directory (FDD) Build (selects from CCMS on-line data bank), (3) Front End Processor Table Build (constructs tables from FDD and Flight Format File data describing FEP data processing), (4) Application Software Configuration (selects application software from the library, updates interpretative code from the FDD, resolves and links all external references), (5) Installation (collects all TCID elements, formats data for CCMS and write data to CDS tapes for tape to disk generation in CCMS or real time CDS to CCMS data transfer) (50KBS).

A Master Math Model Build is also performed to provide simulation capability in the debug and verification process. The Master Model Build selects system models from the model library and creates simulation tables from the FDD. These models interface with CDS resident simulation software for ground software development activities.

The Software debug and verification process is performed in the Ground Software Production Facility (Firing Room 2). Firing Room 2 can be divided into three sets so that multiple TCIDs can be processed concurrent,y. The three sets are flexible in configuration to allow a various number of Consoles and Front End Processors to be used. System configuration ties the Firing Room sets to simulation models residing in the CDS computers via Video Simulation Interfaces.

Following software verification, Software Assurance and Configuration Management provides quality control verification and assures compliance with all effective approved engineering requirements.

*Manager, Ground Software/LPS Operations, RI-KSC

Additionally, this function provides verification of all open items and the generation of Release Notices reflecting the configuration of application software packages prior to the package being used for hardware processing.

SYSTEM BUILD OPERATIONS

The System Build function is the compilation and translation of all application software elements into a controlled applications library resident on CDS. An Applications TCID can be divided into the following categories to support testing and Vehicle processing: (1) Support Equipment Site Activation and Revalidation, (2) STS Element Test, i.e., Orbiter, SRB, ET, Cargo, (3) STS Integrated Processing.

Ground software types included in a peculiar TCID include: (1) (GOAL) Ground Operations Aerospace Language programs, (2) Control Logic sequences, prerequisite and reactive, (3) Display Skeletons, and (4) Test Control Supervisor sequences.

Management of the System Build process is controlled by the System Build Manager (SBM). The System Build Manager is a controlled, interactive, real-time software tool for initiating and executing all TCID build functions. The SBM software was generated to remove manual intervention in the build process, to avoid mistakes and to create a repetitive process. The SBM software assures that (1) The build function being initiated is authorized, (2) Only approved application program revisions are included in the build, (3) Application software compiler release effectivity is correct and (4) Application software console assignments are made. The SBM generates a log for historical record data for inclusion into the Software Release Notice.

The System Build process is controlled by Operating Maintenance Instruction (OMI) V9025. This OMI is broken up into Mini OMI's for individual repetitive execution of standalone tasks. It defines Software Assurance check points and minimizes the opportunity for human error.

This System Build process is depicted in Figure 1.

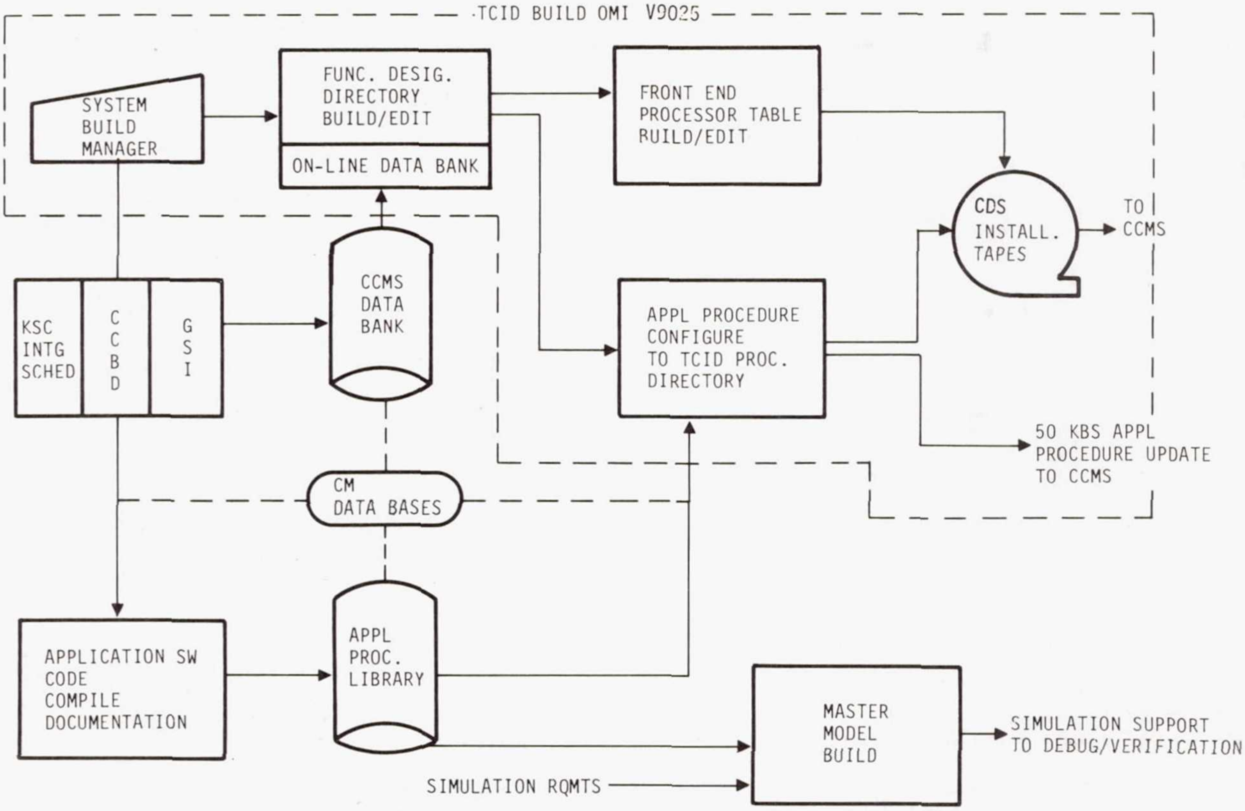
TCID BUILD OMI V9025
SYSTEM BUILD MANAGER
FUNC. DESIG. DIRECTORY BUILD/EDIT
ON-LINE DATA BANK
FRONT END PROCESSOR TABLE BUILD/EDIT
CDS INSTALL. TAPES
TO CCMS
KSC INTG SCHED
C C B D
G S I
CCMS DATA BANK
APPL PROCEDURE CONFIGURE TO TCID PROC. DIRECTORY
50 KBS APPL PROCEDURE UPDATE TO CCMS
CM DATA BASES
APPLICATION SW CODE COMPILE DOCUMENTATION
APPL PROC. LIBRARY
MASTER MODEL BUILD
SIMULATION SUPPORT TO DEBUG/VERIFICATION
SIMULATION RQMTS

Figure 1.- System build.

To meet the challenge of the Operational Era, software development techniques are evolving to allow more automated utilities to control the System Build function to eliminate errors and provide software for multi-flow operations.

A "System Build Requirements Document" concept is presently in development. This document will contain a complete set of hardware and software policies, responsibilities, configuration and interfaces as well as any unique support requirements for each Checkout Control and Monitor Subsystem (CCMS) set.

The System Build Requirements Document will consist of the CCMS data which will control the LPS System Build process. The software configuration (Function Designator Directory Build requirements, table build requirements and application software) and interfaces will be stored on CDS for an Automated TCID Build Process.

Automated utilities will be developed to be initiated by the operator for access to the System Build Requirements Document on CDS to produce a Function Designator Directory Build input deck, a table build input deck and a baseline of application programs to support that TCID.

The intent is to use the System Build Requirements Document to control the LPS System Build process from start to finish by utilizing this Document as well as Automated Software Utilities to fully implement an approved set of requirements. This concept is depicted in Figure 2.

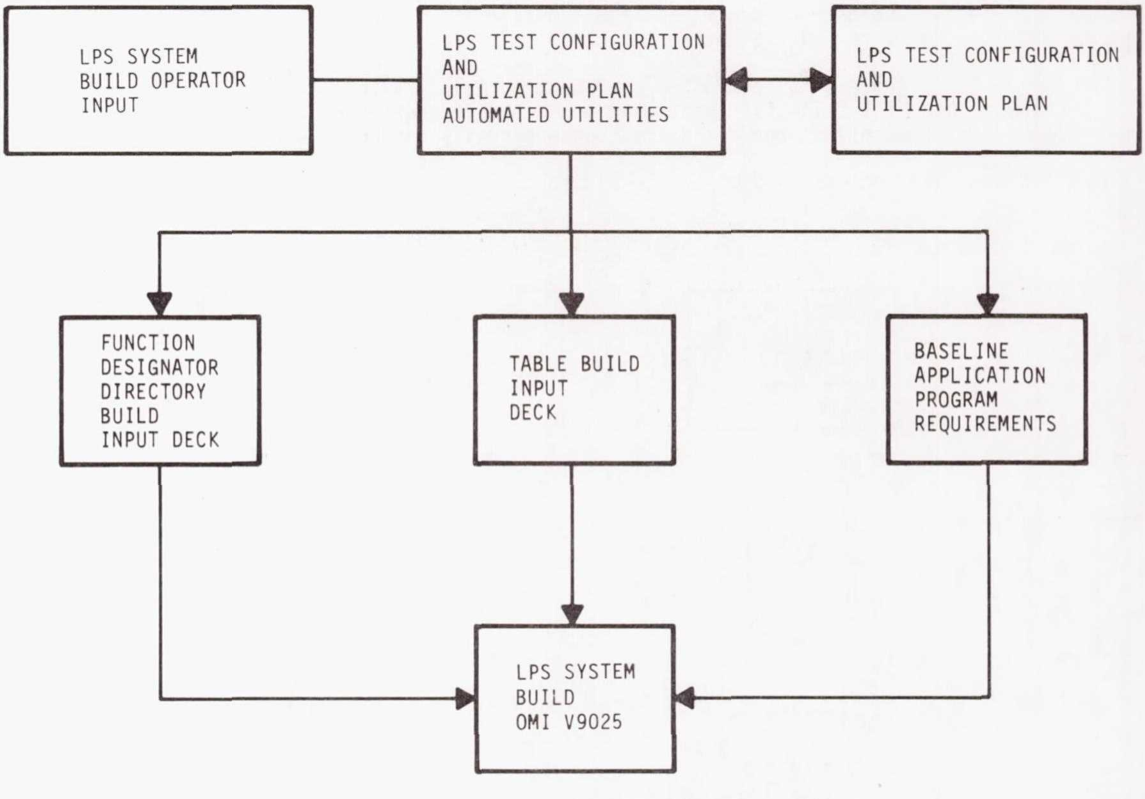
LPS SYSTEM BUILD OPERATOR INPUT
LPS TEST CONFIGURATION AND UTILIZATION PLAN AUTOMATED UTILITIES
LPS TEST CONFIGURATION AND UTILIZATION PLAN
FUNCTION DESIGNATOR DIRECTORY BUILD INPUT DECK
TABLE BUILD INPUT DECK
BASELINE APPLICATION PROGRAM REQUIREMENTS
LPS SYSTEM BUILD OMI V9025

Figure 2.- System build requirement document.

SHUTTLE DATA BANK MANAGEMENT

The Shuttle (CCMS) Data Bank is a large, structured, disk file residing in the Central Data Subsystem. The Data Bank contains all the information on measurements, stimuli, and system parameters needed for the operation of CCMS support software. Among other data, the CCMS Data Bank contains function designator data for each measurement and stimulus of the Shuttle Vehicle. The function designator data consists of the Measurement/Stimulus Identification (MSID) number, name, range, dimensional units and other such characteristics.

The CCMS Data Bank is maintained in three sections on a common disk including a section for Space Shuttle Vehicle (SSV), for Ground Support Equipment (GSE), and one for Cargo/Payloads. KSC is the design agency for the GSE section. JSC, MSFC and Rockwell/Downey are responsible for SSV and Cargo/Payloads.

The use of function designators (FD) is a fundamental GOAL concept that allows/requires the GOAL Programmer to specifically designate the function and destination of external references. Approximately 50,000 FD's reside in the CCMS Data Bank, broken down as follows: (1) Orbiter - 24,000, (2) SSME - 1400, (3) ET - 500, (4) SRB - 1100, and (5) GSE - 23,000. Each FD requires approximately 55 pertinent pieces of technical data for a total of 2.7 million data elements for a single mission.

A Shuttle Data Tape (SDT) supplies the measurement/stimulus data from the design agencies to the LPS for the generation of CCMS Data Bank field data.

The Shuttle Data Tape Processor is a unique software component that forms the repository of Shuttle information required in the CCMS Data Bank under control of the CDS Operating System.

From the raw SDT, the Processor generates a file of card-image updates for the Shuttle associated function designator in the CCMS Data Bank. This update file, along with other required data, will be used to update the CCMS Data Bank. Typical flow of SDT to Data Bank updating process is shown in Figure 3.

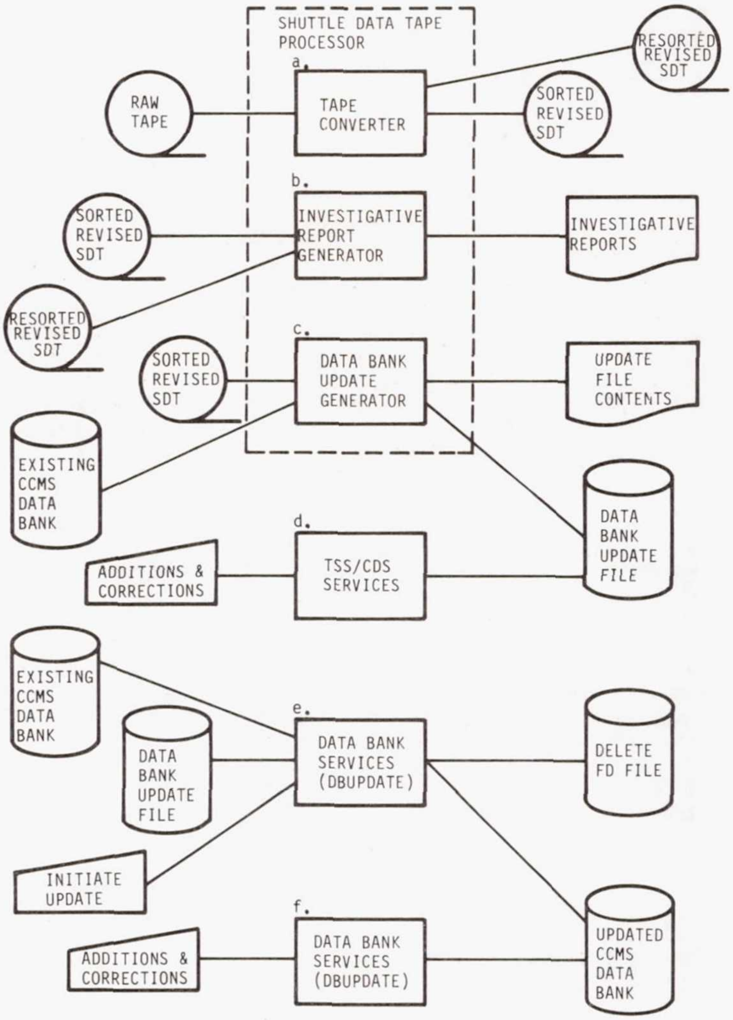

Figure 3.- Typical SDT-to-DB updating process.

Data Bank Services functions are utilized to add, modify, delete, copy transfer, and receive the various types of records in the Data Bank. The Data Bank Update Generator portion of the SDT Processor builds compiler and hardware records according to data link, data type and on-board sources and destination. These records are compared to current records in the Data Bank. Following these comparisons, the Data Bank Generator provides update data for the Data Bank reflecting new or modified data fields from the SDT. Data Bank update change approval flow is shown in Figure 4.

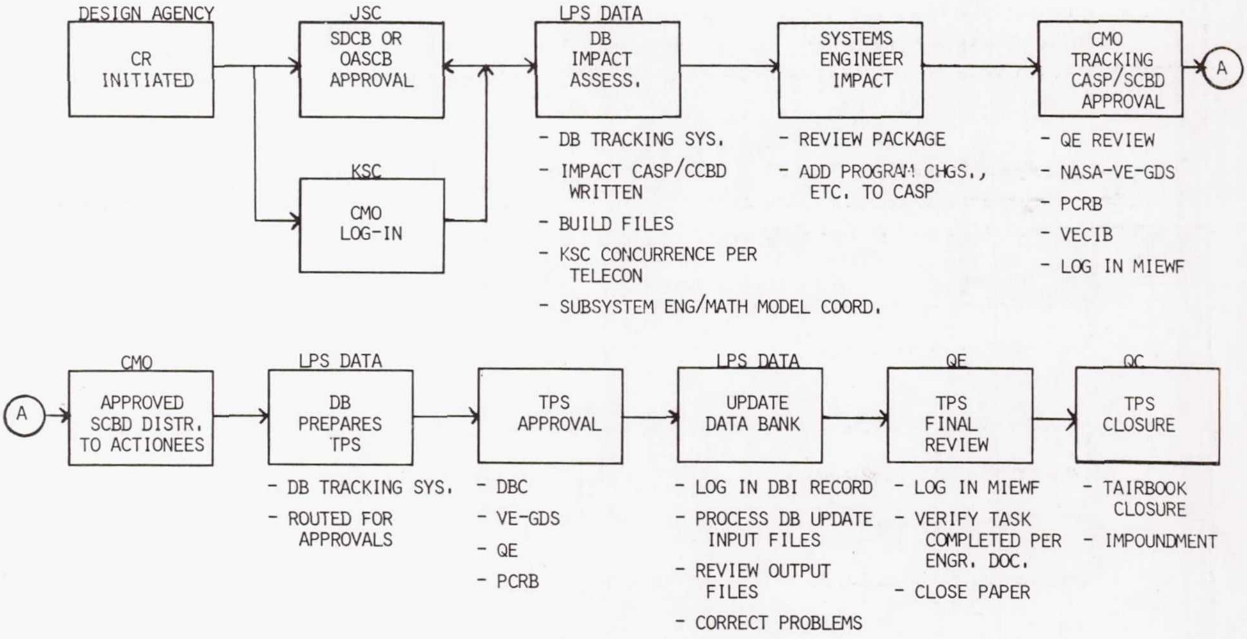

Figure 4.- CCMS data bank update.

FIRING ROOM UTILIZATION

To meet the challenge of the Operational Era, efficient utilization of Firing Room resources will be mandatory to manage multi-flow schedules. Presently, three Firing Rooms are available at KSC for Vehicle processing and software development activities. Firing Room 4 is presently under construction.

Firing Room 2 at KSC has been designated as the Ground Software Production Facility (GSPF). The GSPF has been divided into three sets for maximum resource utilization and allowing multiple software TCIDs to be processed concurrently. Reference Figure 5 for Software Development Configuration.

The three sets are very flexible in order to allow a varying number of Consoles and Front End Processors to be used. The configuration ties the Firing Room sets to simulation models residing in the CDS computers via "Video Simulation" interfaces. Software is executed from the consoles with responses from the system models in the debug/verification process. The system models are controlled from a console in the CDS set. Verification of application programs are performed per approved Software Verification Procedures (SVP's).

Software planning and resource utilization control is managed by software planning and utilization scheduling. The Software Integrated Support Schedule is updated weekly and shows the complete software development cycle from Data Bank, System Build, debug, verification and release to hardware testing.

The STS/Payload 72 Hr/11 Day Operations Integration Schedule is updated via daily meetings to reflect the Day to Day schedule and set utilization. The Software Test Conductor who coordinates the GSPF activity has the authority to make real time schedule adjustments to meet changing real time requirements.

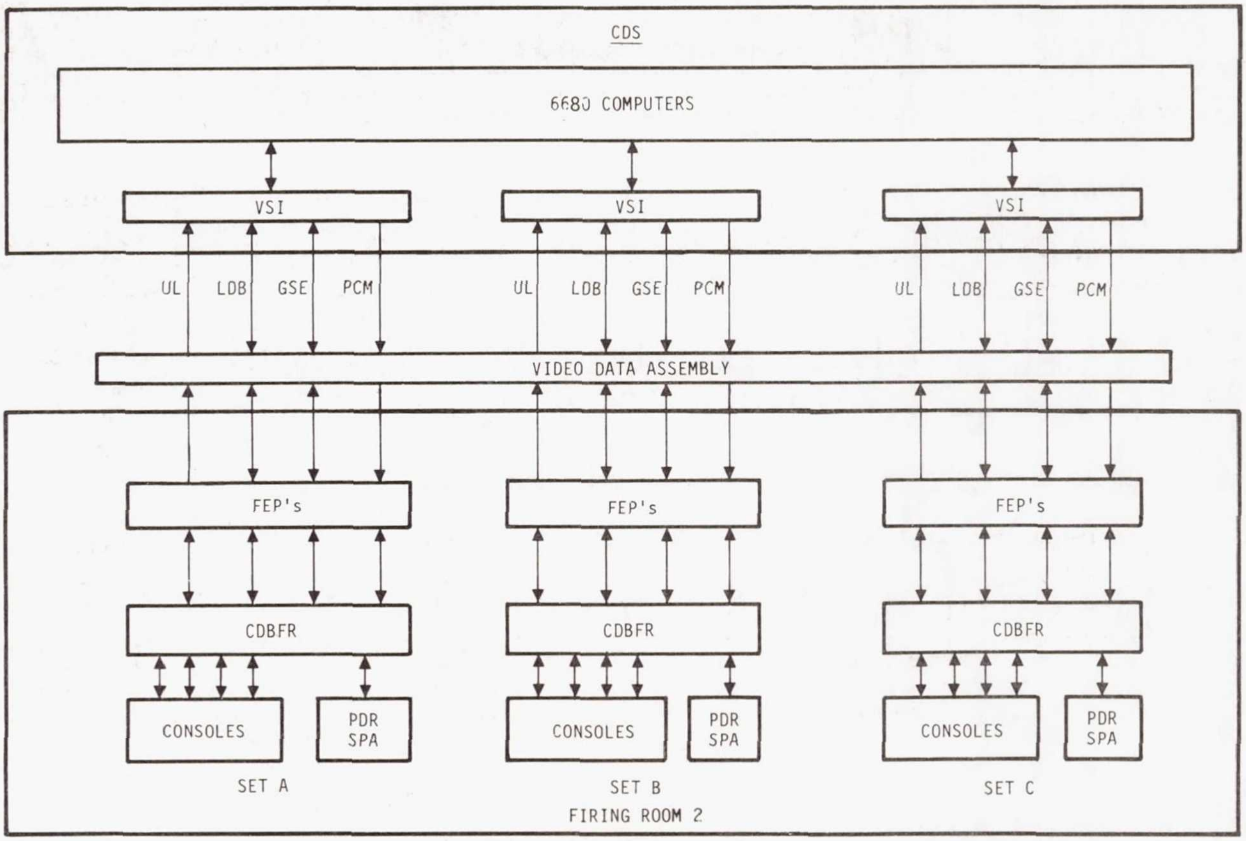

Figure 5.- Software development configuration.

The Software Development/Release process through the Ground Software Production Facility is shown in Figure 6.

Change control, use of automated utilities and configuration management are essential keys to efficient software production operations to meet the challenges of the Operations Era.

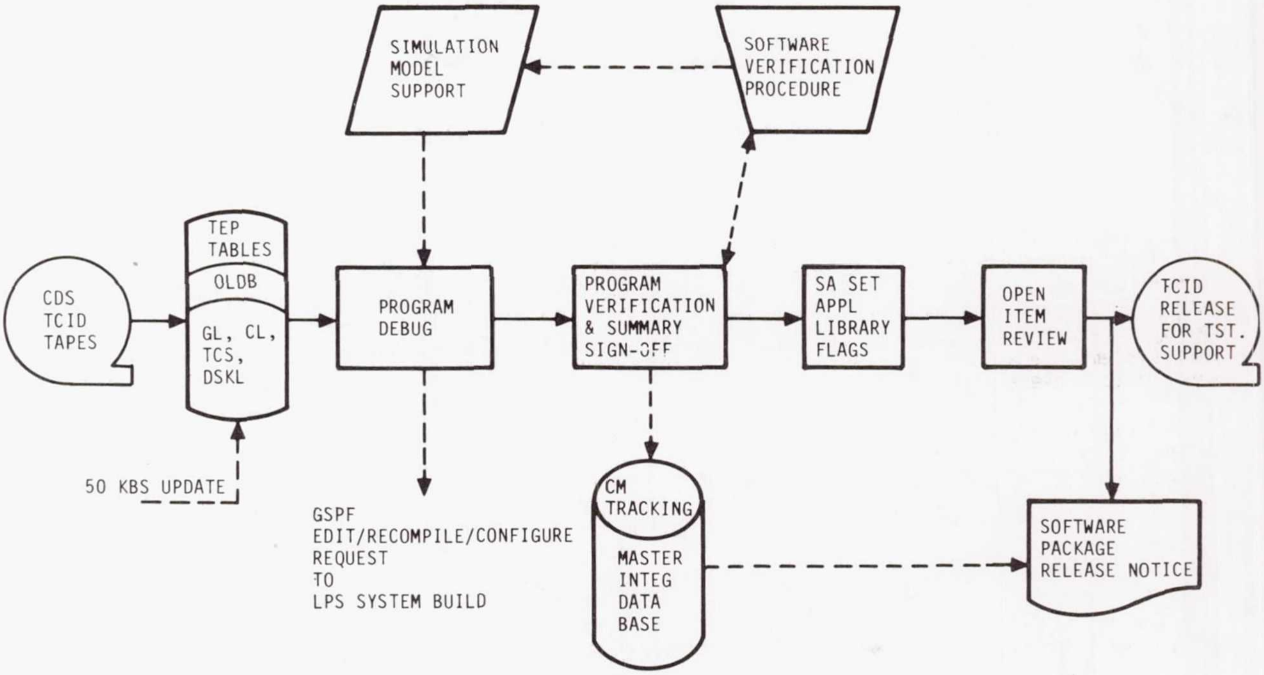
SIMULATION MODEL SUPPORT
SOFTWARE VERIFICATION PROCEDURE
CDS TCID TAPES
TEP TABLES
OLDB
GL, CL, TCS, DSKL
50 KBS UPDATE
PROGRAM DEBUG
GSPF EDIT/RECOMPILE/CONFIGURE REQUEST TO LPS SYSTEM BUILD
PROGRAM VERIFICATION & SUMMARY SIGN-OFF
CM TRACKING
MASTER INTEG DATA BASE
SA SET APPL LIBRARY FLAGS
OPEN ITEM REVIEW
TCID RELEASE FOR TST. SUPPORT
SOFTWARE PACKAGE RELEASE NOTICE

Figure 6.- S/W development/release.