NASA-CR-190714

Ground Systems Development Environment (GSDE) Interface Requirements Analysis: Operations Scenarios

Victor E. Church
John Phillips
Computer Sciences Corporation

Johnson
IN-61-CR
116932
P.75

February 1991

(NASA-CR-190714) GROUND SYSTEMS DEVELOPMENT ENVIRONMENT (GSDE) INTERFACE REQUIREMENTS ANALYSIS: OPERATIONS SCENARIOS Interim Report (Research Inst. for Computing and Information Systems) 75 p

N92-34197

Unclas

G3/61 0116932

Cooperative Agreement NCC 9-16
Research Activity No. SE.34

NASA Johnson Space Center
Mission Operations Directorate
Space Station Ground Systems Division

ricis

Research Institute for Computing and Information Systems
University of Houston-Clear Lake

INTERIM REPORT

The RICIS Concept

The University of Houston-Clear Lake established the Research Institute for Computing and Information Systems (RICIS) in 1986 to encourage the NASA Johnson Space Center (JSC) and local industry to actively support research in the computing and information sciences. As part of this endeavor, UHCL proposed a partnership with JSC to jointly define and manage an integrated program of research in advanced data processing technology needed for JSC's main missions, including administrative, engineering and science responsibilities. JSC agreed and entered into a continuing cooperative agreement with UHCL beginning in May 1986, to jointly plan and execute such research through RICIS. Additionally, under Cooperative Agreement NCC 9-16, computing and educational facilities are shared by the two institutions to conduct the research.

The UHCL/RICIS mission is to conduct, coordinate, and disseminate research and professional level education in computing and information systems to serve the needs of the government, industry, community and academia. RICIS combines resources of UHCL and its gateway affiliates to research and develop materials, prototypes and publications on topics of mutual interest to its sponsors and researchers. Within UHCL, the mission is being implemented through interdisciplinary involvement of faculty and students from each of the four schools: Business and Public Administration, Education, Human Sciences and Humanities, and Natural and Applied Sciences. RICIS also collaborates with industry in a companion program. This program is focused on serving the research and advanced development needs of industry.

Moreover, UHCL established relationships with other universities and research organizations, having common research interests, to provide additional sources of expertise to conduct needed research. For example, UHCL has entered into a special partnership with Texas A&M University to help oversee RICIS research and education programs, while other research organizations are involved via the "gateway" concept.

A major role of RICIS then is to find the best match of sponsors, researchers and research objectives to advance knowledge in the computing and information sciences. RICIS, working jointly with its sponsors, advises on research needs, recommends principals for conducting the research, provides technical and administrative support to coordinate the research and integrates technical results into the goals of UHCL, NASA/JSC and industry.

RICIS Preface

This research was conducted under auspices of the Research Institute for Computing and Information Systems by Computer Sciences Corporation in cooperation with the University of Houston-Clear Lake. The members of the research team were Victor E. Church, John Phillips and Ray Hartenstein from CSC. Mr. Robert E. Coady was CSC program manager for this project during the initial phase. Later, Mr. Ray Hartenstein assumed the role of CSC program manager. Dr. Alfredo Perez-Davila served as RICIS research coordinator.

Funding was provided by the Mission Operations Directorate, NASA/JSC through Cooperative Agreement NCC 9-16 between the NASA Johnson Space Center and the University of Houston-Clear Lake. The NASA research coordinator for this activity was Thomas G. Price of the ADPE and Support Systems Office, Space Station Ground Systems Division, Mission Operations Directorate, NASA/JSC.

CSC/TM-91/6061

Ground Systems Development Environment (GSDE) Interface Requirements Analysis: Operations Scenarios

Prepared for

The University of Houston-Clear Lake
Houston, Texas

by

Computer Sciences Corporation
System Sciences Division
Beltsville, Maryland

under

Subcontract No. 075
RICIS Research Activity No. SE-34
NASA Cooperative Agreement NCC 9-16

February 1991

Prepared by:		Approved by:	
Victor E Church	3/27/91	Ray Hartenstein	3/27/91
V. Church	Date	R. Hartenstein	Date
John S Philips	3/27/91		
J. Philips	Date		

Preface

This report was developed in cooperation with the Research Institute for Computing and Information Sciences at the University of Houston-Clear lake. It was prepared for use by the Johnson Space Center in defining the interface and configuration management (CM) procedures to be used in developing space station ground system software. In working paper form it was used to provide a framework for comments by ground system software developers. The CM procedures and interface functions will be defined on the basis of that feedback. This report provides a record of the analysis at a particular stage, and will not be updated. It presents the content of the working paper with no substantive changes. Comments made to the working paper were not incorporated into this report. A later report will build on the information provided by contractors in response to this information.

Abstract

This report is a preliminary assessment of the functional and data interface requirements of the link between the GSDE GS/SPF (Amdahl) and the Space Station Control Center (SSCC) and Space Station Training Facility (SSTF Integration, Verification, and Test Environments (IVTEs). These interfaces will be involved in ground software development of both the control center and the simulation and training systems. This preliminary report describes our understanding of the configuration management (CM) interface and the expected functional characteristics of the Amdahl-IVTE interface. It presents a set of assumptions and questions that need to be considered and resolved in order to complete the interface functional and data requirements definitions. It includes a listing of information items defined to describe software configuration items in the GSDE CM system. It also includes listings of standard reports of CM information, and of CM-related tools in the GSDE.

Table of Contents

1.0 Introduction

NASA has directed that the Amdahl computer also refered to as the Ground Software /Software Production Facility, GS/SPF, located in building 46, is to provide a central Configuration Management (CM) repository as implemented by the Software Support Environment (SSE) contractor. The SSE includes support tools necessary to perform software verification and test under tight CM control. Since the actual verification and test processes will take place largely in two Integration, Verification and Test Environments (IVTEs) provided by the Mission Systems Contract (MSC) and Training Systems Contract (TSC) contractors, the operational interfaces between the IVTEs and the Amdahl are of critical importance.

The MSC and TSC contractors will share the GS/SPF Amdahl computer as part of the Ground Systems Development Envirionment (GSDE). In view of this shared resource it is important to identify commonality between the two IVTEs, as well as the differences between them.

1.1 Purpose

This report is part of a study of the Amdahl-to-IVTE interface. The study is designed to identify the functional requirements and data transfer requirements of that interface. The implementation of that interface is expected to involve the MSC, TSC, and SSE contractors. A major goal of this study is to provide information to assist NASA in coordinating that implementation.

This document is intended to serve as a catalyst in the process of gathering the information needed to define the interface software requirements and in specifying the ICDs between the interface software and the existing (or planned) IVTE operational software. Therefore it both presents information concerning our understanding of the SSE and IVTE functions; and, more importantly, requests information concerning the anticipated application of the CM tools, processes, and methodologies in the integration, verification, and testing of software for the SSCC and the SSTF.

The primary vehicle used is the "scenario" each of which is intended to present a detailed walk-thru of a "typical" verification and test process. Many such scenarios were studied and a reduction process was applied in order to minimize the effort required while encompassing the needed information. Thus some scenarios were combined with others while some were deleted if they did not add information. The three scenarios that are presented have no special significance other than that they seemed to capture the required information. We are open to and would welcome suggestions for any improvement in reaching our goal.

Specifically, we request responses to the questions that are presented throughout the scenarios in section 4, and an assessment of the accuracy of assumptions we have made. The questions and assumptions are collected at the end of section 4.

1.2 Scope

This task is an important step in the process of assuring that the Amdahl to IVTE interface is as efficient and concise as possible and that it provides a productive, CM-compliant "connection" for the SSCC and SSTF software developers. Though it was occasionally necessary to reference information about the development of software in the scenarios they are not intended to address nor be authoritative in this area. The scope of this task is presently confined to the interface software required to connect the Amdahl and the IVTE's. Further, it is not the intent nor within the scope of this document to dictate the methodologies that will be used to develop or test software but rather to collect specific information. Corrections and/or suggestions regarding the accuracy/efficacy of these scenarios would be welcomed, particularly if those corrections impact the information requested.

1.3 Organization

Section 2 of this interim report describes the structure of the CM system within the Amdahl. Section 3 describes some of the basic assumptions we have made and issues we have identified in analyzing the Amdahl-IVTE interface. Section 4 presents three scenarios describing how the interface might operate during development and test. Section 5 consists of tables describing the data required by the CM process. Additional information about CM reports and tools identified during this study is provided in appendices .

2.0 Configuration Management within the GS/SPF (Amdahl)

This section describes, in general, the CM system that will be provided by the SSE as it pertains to providing configuration management for the integration and testing of software. This section will be broken down into several sub-sections that describe the different aspects of the CM system.

2.1 General Configuration Management Description

The CM system provided to the GSDE SPF Amdahl by the SSE is an Oracle based system of tools and a tracking data base. This system has some (but not all) interfaces needed to collect data throughout the life cycle of any software. In addition, the system has tools that allow the developer or tester to construct information reports that will be used throughout the software life cycle.

For the purpose of description and discussion there are three main divisions; the CM Fields, the CM Reports, and the CM Tools . Each of these will be discussed in the following sub-sections.

2.1.1 Configuration Management Fields

Configuration Management Fields are those information fields that make up part of the SSE CM Oracle database. The fields in the table located in Section 5.0 following the scenarios will be fields of interest for the scenarios that are part of this document. The listing here does not represent all the CM fields that are available. Many more fields are listed in the tables that include the CM tools and the CM reports fields.

The fields listing is made up of four columns. The first column is an arbitrary field number that is used in aligning the fields listed in this table with the CM Data Item column in the scenarios. The second column is the name of the field as defined in the SSE requirements. The third field is a best guess at whether or not the data values for the associated field will be generated automatically or will be a manual entry. This entry information is not a reflection of the requirements for the SSE provided CM but rather an assumption on the author's part in trying to determine the amount of manual entry associated with the CM process. As part of the comments of the readers it is encouraged that comments about the generation of the data values for these fields be expressed. The fourth column is blank. This column is for noting "who" is the responsible party for annotating the data value for any field. By "who" it is meant "who organizationally" and where this person is located (either in the IVTE or working through the Amdahl). It is requested that the readers fill in this column to the best of their ability. It is hoped that this information will shed some light on the problem of getting CM information from the IVTE to the Amdahl CM.

2.1.2 Configuration Management Reports

The SSE CM Oracle data base includes the capability for authorized users to generate many pre-defined reports as well as create custom reports from the information within the CM. All these reports may be edited, modified, have information added or deleted, or logged. A full listing of the reports and their contenets are in Appendix A.

2.1.3 Configuration Management Tools

Appendix B is a listing of the CM Tools that are provided to the user when SSE OI 6.0 is delivered. The list is made up of two columns. The first column is the name of the tool and the second is the capabilities of the tools. Of special importance is the fact that these tools are linked to the CM records so that by using the tools, much of the CM information that will be required can be generated with the tools and "passed" to the CM records with little effort.

3.0 GS/SPF (Amdahl)-IVTE Interface analysis

This section uses the technique of operations scenarios to describe a possible interface architecture between the GS/SPF Amdahl and the SSTF and SSCC IVTEs. Several different operations involving inter-environment communications are described and analyzed. These scenarios are based on some assumptions about the use of the IVTEs, and lead to some questions about the functional and logical interfaces between the Amdahl and the IVTEs.

All of these assumptions and questions are reasonable topics for further discussion and analysis. In many cases the answers may simply not be determined yet; other issues may depend on the completion of the OADP procurement. The intent of this section is to identify issues and assumptions that must be clarified to ensure that the interface operates as desired.

Three scenarios are presented. The first involves using the IVTE to compile source files which are being promoted to "ready for test" status. The second involves creating executable objects in the IVTE. The third involves integration testing of CIs in the IVTE.

The scenarios are used to provide a *framework* for discussion of issues involving the GS/SPF-IVTE interface. It is *not* our intent to prescribe these scenarios as the only way of developing software; rather, we wish to highlight interface considerations that need to be addressed if our assumptions are reasonably accurate. Some of the questions and issues may already be resolved; this study may provide a convenient framework for recording those resolutions and coordinating them between the SSCC and SSTF teams.

3.1 Basic assumptions

The basic picture of the Amdahl-to-IVTE interface that we have assumed is shown in figure 1. Development personnel interact with the Amdahl to effect configuration management (CM) of files, and of their attributes and relationships recorded in the CM system. Software configuration items (CIs) are downloaded from the Amdahl with appropriate processing instructions (command scripts). Processing (e.g., integration testing) occurs in the IVTE, including interactions with IT&V personnel. There is no direct interactive (i.e., workstation-based) link between the IVTE and the GS/SPF.

Products and status are returned to the Amdahl after processing. Products generated on the IVTE (e.g., object code) may be retained there for further use as well as being uploaded to the Amdahl.

Figure 1. GS/SPF (Amdahl) to IVTE Interface

3.2 Discussion of major topics

There are several themes that run through the analysis of Amdahl-IVTE interface functions and specifications. These themes are discussed below. In general they involve questions of system engineering of the interface (i.e., what functions will be performed in what subsystems?). The fact that the Amdahl must provide interface and functional support for two different IVTEs is a factor in partitioning functionality, and is important for the interface definition process. Most of the issues described below are addressed in the three scenarios.

These topics are the core of this interface analysis, which involves the specification of Amdahl-IVTE functional and data interfaces.

3.2.1 Amdahl-IVTE functional interface architecture

There are several interrelated issues involving the support that the Amdahl and the IVTEs provide to each other. These issues are central to the analysis and specification of functional interfaces. The basic question is how the necessary functionality is to be partitioned among the Amdahl, the IVTEs, and the users of the three systems (i.e., the level of manual operation required). Some of the interface functionality can be provided by commercial software, but every functional allocation has implications for other support elements (automated or manual). These issues include:

A) What handshaking or message-response protocol will exist between the Amdahl and the IVTEs? Will the protocol involve anything more complex than datagram-like service? (That is, will the protocol consist of exchanges of messages and files with no direct acknowledgement beyond what the supporting network provides?)

B) Will there be application-to-application interfaces between the Amdahl and the IVTEs, such as would be required for directory inquiry service?

C) What is the partitioning of script-processing functions between the two types of systems? Will the IVTE host know anything about the Amdahl, or will it simply generate products (e.g., load modules, test results) that users or Amdahl software must interpret? Will the Amdahl download files (e.g., target loads) directly into target processors, or will some software in the IVTE host perform a store-and-forward function?

These questions are bound up in the specific areas outlined below.

3.2.2 IVTE interface functional architecture

The software in the IVTE that supports the interface with the Amdahl has several functions, including communications and file transfer, directory services, and internal file distribution. The functionality of this interface could be centralized in an IVTE host, or could be more or less distributed over all the platforms in the IVTE. This question is somewhat linked to the degree of automation desired, and to the partitioning of functionality between the Amdahl and the IVTEs.

3.2.3 Security

To ensure the integrity of information stored in the IVTE, provisions must be made to restrict access and particularly to control incoming data. There are several possible policies for security on this communications link:

A) Nothing is sent to the IVTE from the development environment (including workstations) except files that are first placed under CM on the Amdahl. This would guarantee that everything sent to the IVTE has been reviewed, and is saved for later analysis in the event of problems. This policy would block access to the IVTE via workstation (i.e., interactive use of the IVTE from development workstations). It would also control the use of the IVTE for development compilation and unit test, since only controlled files would be permitted.

B) No free-form interactions or command files are permitted, but controlled modifications to CM-controlled objects (e.g., command scripts) are acceptable. That is, modifications like supplying parameters or changing process options (e.g., compiler listing options) could be made to predefined command scripts. The Amdahl retains control of the modification process (e.g., with forms-completion software).

C) Free-form interactions (such as logging in to the IVTE from the SPE) are permitted from secure workstations. This could be achieved by session control in the Amdahl, or by security in the IVTE which could bypass the Amdahl.

3.2.4 File and namespace distribution

Most scenarios which seem likely involve some storage of controlled files (e.g., object files, data, executable images) in the IVTE as well as in the Amdahl. There are several ways that the IVTE and Amdahl file systems can be interrelated. (Note that only those files which are duplicated in the IVTE are of concern. The overall IVTE file system is not necessarily integrated with the Amdahl.)

If there is in fact CM-administered storage in the IVTE, then command scripts constructed on the Amdahl (by an Amdahl-based TBU-like facility) must have generic names instead of actual file specifications. The script processor must be able to bind file locations to names to form commands that a host OS can process. The binding of file names to locations can be performed early or late, depending on how tightly the Amdahl and IVTE file systems are integrated. Some possibilities are:

A) The Amdahl maintains a copy of the IVTE directory of duplicate files. Some process running in the IVTE host provides updates to the Amdahl (perhaps on a schedule, or by request, or upon changes to the "duplicated file" directory). A process on the Amdahl uses these updates to maintain its directory.

A') An alternative update procedure (with the same duplicated directory mechanism) restricts changes in the IVTE to those which are pre-approved on the Amdahl. The Amdahl directory is kept current by tracking approved file changes.

B) The Amdahl has the capability to inquire about a specific file on the IVTE. Rather than maintaining a directory, the Amdahl gets current information as it needs it. A process on the IVTE host maintains the directory of duplicated files and responds to inquiries.

C) The Amdahl has no way of interrogating the IVTE, and makes assumptions based on CM records about what files can be found in the IVTE. If an expected file is in fact not there, the IVTE has the capability of requesting that specific files be dowloaded at need. The Amdahl includes software that can respond to such a request.

D) There is no file dialog at all. If a file that is expected to be in the IVTE is not there, the operation that needed the file is cancelled. Users on the Amdahl may maintain their own lists (e.g., IVTE directory listings) of files and locations.

3.2.5 File integrity

Some types of files are particularly subject to editing during the process of development and test. Test data, test scripts, test utilities, and similar resources are often modified during testing to support more complete characterization of anomalies. There must be some mechanisms to ensure that the correct versions of IVTE-resident files are available as expected. The SSE will provide mechanisms for file comparison and checksum determination of files. How will those mechanisms, or their equivalent, be employed to ensure the integrity of the acceptance test process? Where will the file verification take place: in the IVTE host, on specific platforms, or on the Amdahl?

3.2.6 CM data preparation

There are two sets of circumstances wherein activities in the IVTE lead to changes in CM data on the Amdahl. The first case is when controlled objects (e.g., executable images) are created in the IVTE and uploaded to the Amdahl for configuration control. There must be descriptive CM information provided along with the uploaded objects. The second case is when some event in the IVTE (e.g., a test being passed) needs to be reflected in the CM database. There are several possible mechanisms for this information flow:

A) There is no automated processing. A user logged in to the Amdahl enters data via electronic forms to create and/or update CM records. The information for these updates is provided (offline, probably on paper) from users in the IVTE.

B) There is automated processing on the Amdahl of IVTE products to generate CM data. The IVTE simply uploads raw data (e.g., compiler output listings with the compiler name and version in the header) to the Amdahl. All extraction and processing occurs in the Amdahl.

C) Software in the IVTE performs data analysis and extraction of CM data. The IVTE packages the CM information and uploads it, along with any products (e.g., test output) to the Amdahl. The Amdahl reformats the data, if necessary, and generates CM updates.

4.0 Operations Scenarios

This section presents three scenarios that describe the interface between the Amdahl and an IVTE. The scenarios are deliberately not specific as to which IVTE they relate to. Assumptions and questions are embedded throughout the scenarios; these questions and assumptions, along with identified functions and data flows, are also collected in subsection 4.6.

4.1 Purpose of the scenarios

These scenarios were developed to help characterize the interface between the GSDE Amdahl and the SSTF and SSCC IVTEs. Specifically, they are intended to identify the types of data transfers and protocols that must exist to support CM on the Amdahl while target-based compilation, linking, and testing is performed in the IVTEs. These scenarios present possible sequences of events, and identify the interface functions and data traffic implied by these events.

The scenarios are based on a number of assumptions, both major and minor. These are described either in footnotes or as specific questions interspersed throughout the steps of the scenarios. We invite comments on any of the assumptions (including those which we may have failed to state!); we request responses to the specific questions which are highlighted through the scenarios.

Throughout the scenarios we have identified information which is extracted from, or required by, the Amdahl-based CM system. Collectively this information will form the basis for an interface requirements definition. If the transfer of any of this information between the Amdahl and the IVTEs is inconsistent with the designs of the IVTEs, we request that such inconsistencies be noted. We also request assessment of the assumptions and responses to the questions collected and numbered in subsection 4.6.

4.2 Scenario format

Each scenario is presented as a sequence of specific steps, in tabular form, with discussion and commentary interspersed. Comments that are specific to a single step of the sequence are shown in smaller type following the step they discuss. Comments, questions, and assumptions that refer to the scenario as a whole are in normal type. Explanatory notes that are not critical to the operational flow are presented as footnotes.

The first four columns of the scenario tables are self-explanatory. The **Int/f functions** column identifies functions to be performed by the interface software on the Amdahl or the IVTE. Steps that do not appear to involve the Amdahl or IVTE interface software--

that is, that are completely contained in one environment or the other-- are marked N/A (not applicable).

The **CM information** column identifies the information supplied by or provided to the Amdahl-based CM system. Most of this information is in the form of references to the table in section 5. The numbers in parentheses are data items provided by the CM system to the interface; the data items that are reported back to the Amdahl CM system are not in parentheses.

The **F/N** column is used for footnotes.

4.3 Scenario 1: Target-based compilation

Compilation of operational software will occur both in the SPEs (software production environments) and in the IVTEs. While most software will be compiled using "compile engines" in the SPEs, there will be occasions to use the resources of the IVTE. In such instances there will be specific procedures for downloading the files to compile and for retrieving the output. The products will be annotated for configuration management (CM) purposes, and uploaded to the Amdahl for controlled storage. (Working copies of object files may be left in or transferred back to the IVTE for ready availability).

Both Link and Loral, in presentations concerning planned use of the GSDE, have indicated that code development at least through unit testing will be performed in the SPEs. Compile engines for all target platforms will be available in the SPEs. However, for reasons including load-sharing and more rigorous control of environments, we assume that the IVTE platforms will also be used for some compilation. This scenario assumes that SSE-provided or compatible tools will be used to support code development, including CM, within the SPEs. Support for comparable activities in the IVTE, including annotation of products for Amdahl-based CM, must be accomplished with IVTE-based tools and/or procedures.

Scenario context

The subject of the scenario is the target-based compilation of two CSCIs. One, named **CtrlDev**, is part of the operational software for the target system and so is under Amdahl-based CM control. It is a workstation-based application which interacts with specialized hardware devices and data links. The other, **T-Driver**, is a test tool (also under CM) that simulates telemetry streams from pre-recorded data. It is designed to execute on the IVTE host computer. Both configuration items (CIs) have been compiled, linked, and unit-tested in the SPE. However, since the compilation facilities in the SPEs are not as tighty constrained or controlled as the IVTE facilities, compilation to produce CM-controlled object files is performed in the IVTE.

Figure 2 shows the basic structure of this scenario.

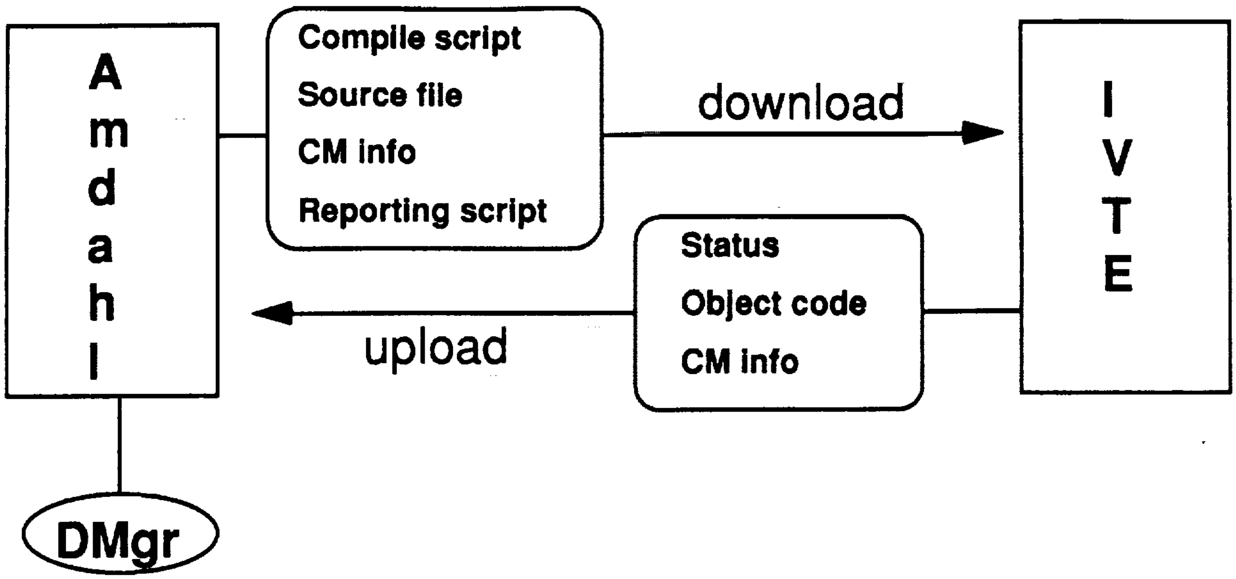

Figure 2. Target compilation scenario

Step 1 Assemble compilation items

The starting point of this scenario assumes the availability of all source and library files needed for compilation. The source code has been placed under CM so that it can be downloaded to the IVTE for the purpose of generating a controlled product. The development manager (DMgr) generates a target compile command script to download necessary files, perform the compilations, collect CM-related information, and upload the object files.

Step	Operator	Element	Process Narrative	Int/f functions	CM information	F/N
1	**Devel. manager (DMgr)**	**Command Script**	**Generates target-specific set of commands to perform target compiles**	**see lower-level steps**	**(66,125,124,134, 133,67,69,120, 124,135,160,162, 163,167)**	1
1.1A	DMgr	Compile instructions	Uses static analysis tools to identify all of the associated libraries and files needed for compilation	na	(8,66,67,68,69,110, 118,135,146,147, 153, 155,162,163, 164,166,167)	

Since the code has been compiled before, this information may already be available in the CM system for reuse.

1 These steps do not directly involve the interface with the IVTE; they are included to establish the context for later operations in the scenario.

Step	Operator	Element	Process Narrative	Int/f functions	CM information	F/N
1.1B	DMgr	Compile instructions	Retrieves stored compilation command scripts	na	CM Tools Provide this Information	
1.2	DMgr	Command scripts	Assembles composite command script to download and process all necessary files	na	build information (63,64,65,66,67, 134,148,138)	2
1.3	DMgr	Target-specific command scripts	Uses target specific translators to generate command scripts for target processors	na	Configuration ID of Command Script	

The SSE System Project is developing a generic capability to instruct an IVTE to perform operations such as compilation. The SSE SP will also provide target-unique translators for different platforms, but expects that the *specifications* for such targets will be provided by IVTE users via CR.

Question: what will the command script language look like? Will it simply be the job control languages of the affected computers, or will there be a special purpose language processor, common to many or all computers in the GSDE and the IVTEs?

1.4	DMgr	CI location data	Identifies the resources that are already resident in the IVTE	TBD	unique CI IDs, location data (118,120)	

For source code compilation processing, it may be adequate to assume that all needed files will be available on demand. If any files are *not* found, the process is simply cancelled.

Question: How will a user on the Amdahl (e.g., the DMgr) determine what files (and what versions of files) are available in the IVTE? Or, if the process is automated, how will the Amdahl make such determinations?

The resource inquiry process may involve a dialog with the IVTE, or examination of a dataset on the Amdahl. (If the latter is the the case, there must be a mechanism for creating and updating that directory dataset.)

2 The command scripts may contain instructions to be processed by several different computers, for example: the Amdahl to retrieve and download files, the IVTE host to receive downloaded files and to retrieve IVTE-resident files, and workstations to compile source code.

Step	Operator	Element	Process Narrative	Int/f functions	CM information	F/N
1.5	DMgr	Command script	Modifes the command script to point to IVTE-resident files	directory services	(Configuration of new script)	
1.5.1	DMgr	Command script	Tailors the script to identify specific resources	na	na	

The DMgr edits the command lists to specify particular resources in the IVTE (e.g., the particular workstations to be used, files that are already IVTE-resident).

Step	Operator	Element	Process Narrative	Int/f functions	CM information	F/N
1.5.2	DMgr	Command script	Includes any necessary security access data to retrieve files in either environment	na	(173,174)	3

This scenario assumes that for security reasons, the download and retrieval mechanisms will only operate on controlled files with proper access authorization.

The requirement for security in the IVTEs may complicate the question of editing command lists for specific operational conditions (e.g., directing files to a chosen target platform per system scheduling). Uncontrolled command script editing could pose a serious security risk. One possible solution is to parameterize the command scripts, perhaps with a forms-filling utility to support the editing of specified parameters.

Question: what mechanism will be used for tailoring command scripts that are targeted to the IVTE?

Step 2 Download and retrieve files

The DMgr activates the command script, causing the desired files to be retrieved and downloaded (if they are on the Amdahl) or retrieved in the IVTE.

3 Access authorization is of concern to this interface study because of the need to direct the use of files in the IVTE, from the Amdahl. This scenario assumes a relatively simple security system, e.g., RACF.

Step	Operator	Element	Process Narrative	Int/f functions	CM information	F/N
2.0	**DMgr**	**Command script**	**Executes script on Amdahl and on IVTE**	**see lower-level steps**		
2.1	Amdahl script processor	Command script	Establishes communications link with IVTE host	job-ID support	transaction ID (job number)	4
2.2	Amdahl-SP	download script	Downloads command script for IVTE and files for script to operate on	File transfer	(63,64,65,66,69)	

The actual file transfer protocol is immaterial to this discussion. The appplications on either end of the transfer are the focus of this analysis.

Step	Operator	Element	Process Narrative	Int/f functions	CM information	F/N
2.3	IVTE script processor	file process script	Retrieves (or points to) IVTE-based files for use in compilation	na	na	
2.4	IVTE-SP	IVTE-based files	Verifies that IVTE-resident files are identical to the controlled versions stored on the Amdahl	file compar-ison	(63,64,65,66,69)	

File verification may be a part of the security access procedures. This scenario assumes that the SSE-provided file verification procedures are used in both the Amdahl and the IVTE.

The IVTE provides a response to the Amdahl that the requested operation is underway. There is no need for the communications link to remain open during the processing on the target platforms (although that may be the simplest approach). There *is* need for some form of acknowledgement of communications.

4 For purposes of handshaking and closure of CM actions, a transaction ID or job number is assigned to the process. This ID is used for confirmation, and to correctly link returned products with downloaded commands.

Step	Operator	Element	Process Narrative	Int/f functions	CM information	F/N
2.5	IVTE-SP	download status	Confirms successful receipt of all files and retrieval of local files	IVTE int/f Ack function	transaction ID (job number)	
2.6	Amdahl-SP	process status	Places transaction ID in "pending completion" status to await products	Amdahl int/f job manager	(66,67,68,69,135, 148,153,154,155, 156,157,158,160, 162,163,166,169, 170,172)	

One side of the interface or the other, or both, need to track what jobs are in progress.This scenario assumes that both ends of the interface perform some transaction (job) management functions.

Question: what mechanism(s) will be used to provide traceability between processing requests and output? Will the mechanism be manual, Amdahl-based, IVTE-based, or some combination ot these?

Step 3 Perform compilations

The script processor on the IVTE directs the appropriate jobstreams to the target platforms for **CtrlDev** and **T-Driver**. The former is downloaded to an IVTE workstation serving as a compile and link engine; the latter is processed on the IVTE host. Status and products are returned to the Amdahl upon completion.

3.0	**IVTE**	**Monitor and T-Drive**	**Compile source code on specified platforms**	**see lower-level steps**	**na**	
3.1	IVTE-SP	**Monitor** job	Transmits command script and files to target platform	file distribution	na	
3.1.1	IVTE-SP	libraries files	Downloads interface libraries and files as required	network int/f	na	
3.1.2	workstation (W/S) platform	Command script	Receives and processes command script	network int/f	na	5

The compile environment for **CtrlDev** includes tools (e.g., Ada compiler), system libraries, and custom shared libraries that are specific to this set of CIs.

5 This scenario assumes that any compile-engine platform will have the capability of receiving and executing command scripts via communications links.

Step	Operator	Element	Process Narrative	Int/f functions	CM information	F/N
3.2	W/S platform	**CtrlDev** job	Compiles source and creates object, listing, and status information	na	66,67,68,69,135, 148,153,154,155, 156,157,158,160, 162,163,166,169, 170,172	
3.3	W/S platform	**CtrlDev** products	Uplinks product and status to IVTE host	network int/f	na	

The compilation of this CSCI is successful. Status therefore includes not only the fact of success but identification (name and version) of each system tool used in creating the product (e.g., compiler, system library).

Question: certain information must accompany the object code when it is uploaded to the Amdahl for CM. Where does that information get extracted, formatted, and linked with the object code files? This scenario assumes a distribution of functions among the target platforms, the IVTE host, and the Amdahl.

Step	Operator	Element	Process Narrative	Int/f functions	CM information	F/N
3.4	IVTE host	Products and status	Receives output and spools it for uplink processing	network int/f	na	6
3.5	IVTE host	**T-Driver** job	Executes command script, compiling source, collecting info, and reporting job status	na	66,67,68,69,135, 148,153,154,155, 156,157,158,160, 162,163,166,169, 170,172	

The compilation of this CI fails due to a mismatch with one of the IVTE-resident libraries. The products therefore include the output that identifies the reason for failure, and the completion status that will close the CM operation.

Step	Operator	Element	Process Narrative	Int/f functions	CM information	F/N
3.6	IVTE-SP	products and status	Assembles job completion and CM records needed to complete the operations	na	na	

Step 4 Upload products

The IVTE host packages the products of the operations for uploading to the Amdahl. The session ID acts as a reference. CM-required information is formatted for use in creating new controlled files and annotating old ones.

6 The IVTE host acts as the interface between the IVTE and the Amdahl.

Step	Operator	Element	Process Narrative	Int/f functions	CM information	F/N
4.0	**IVTE script processor**	**products and status**	**Package and upload products**	**see lower-level steps**	**na**	
4.1	IVTE-SP		Establishes or resumes communications link with Amdahl	IVTE int/f	na	7

The IVTE host interface software determines that the IVTE processing of the transcation has been completed, and initiates efforts to complete the process by uploading results. The communications session may still be in place, or may have to be reinstated.

Question: can transactions be initiated from the IVTE? Is the process automatic or is operator intervention required?

4.2	IVTE-SP	Job Status	Assembles and formats CM records to support CM operations on the Amdahl	CM data formatter	63,64,65	8

The IVTE host performs target-unique data extraction and converts the information into a standard format that can be processed on the Amdahl

4.3	IVTE-SP	Job products	Packages CM data, products, Job ID info	na	na	
4.4	IVTE-sp	Product packages	Uploads packaged information to Amdahl	IVTE int/f	na	

Step 5 Process and store products

The Amdahl receives the products, directs them as appropriate, and submits information to the CM system to conclude the scenario.

7 The IVTE may have the capability to initiate upload operations, or the interactions may be controlled from the Amdahl.

8 See discussion at the end of this section, on distribution of functionality

Step	Operator	Element	Process Narrative	Int/f functions	CM information	F/N
5.0	**Amdahl-SP**	**Products**	**Performs recording and distribution**	**see lower-level steps**	**na**	
5.1	Amdahl-SP	Command script	Retrieves suspended script based on transaction ID returned with uploaded products (see step 2.6)	na	Cm assigns transaction ID	
5.2	Amdahl-SP	CM records	Processes CM records for submission to the CM system	CM formatter	na	9
5.3	Amdahl-SP	CM products	Submits **CtrlDev** object files as derivative CIs for CM	na	Cm assigns Configuration ID to object code	
5.4	Amdahl-SP	Process products	Distributes non-controlled products (e.g., listing files) as appropriate	na	na	

4.4 Scenario 2: Target-based load building

Creation of test-ready target loads will occur both in the SPEs (software production environments) and in the IVTEs. When the resources of the IVTE are used (e.g., when a particular set of IVTE-only hardware and software is required), the process involves transferring necessary source and/or object files to the IVTE and processing them on target systems. The products are annotated for configuration management (CM) purposes, and uploaded to the Amdahl for controlled storage. (Working copies may be left in or transferred back to the IVTE for ready availability).

Scenario context

The subject of the scenario is the target-load building of two CSCIs. One, named **Monitor**, is part of the operational software for the target system and is under Amdahl-based CM. It is a workstation-based application which interacts with other workstations and data links. The other, **T-Driver**, is a test tool (also under CM) that simulates telemetry streams from pre-recorded data. It is designed to execute on the IVTE host computer. Both configuration items (CIs) have been compiled, linked, and unit-tested in the SPE; controlled object code exists for both CSCIs. However, since the object libraries in the SPE are not identical to the operational object libraries (e.g., they contain

9 The formatting tools on the Amdahl and the IVTE host are complementary; the functionality could be distributed to either system

additional stubs and test modules that are not part of the operational system), the CSCIs will be rebuilt in the IVTE for integration testing.

Figure 3 illustrates this scenario.

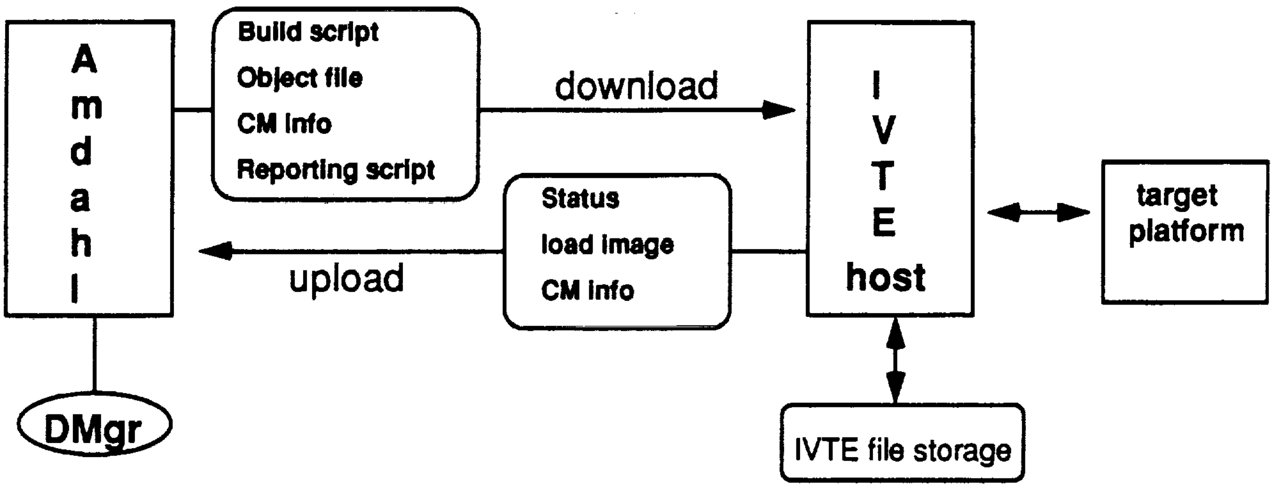

Figure 3. Target-load building

Step 1 Assemble build items

The starting point of this scenario is the availability of all object files needed to build the target loads. The source code is under CM following system test, and has been compiled to generate object code which is also placed under CM. The development manager (DMgr) generates a target build command script to download necessary files, perform the builds, collect CM-related information, and upload the products.

Step	Operator	Element	Process Narrative	Int/f functions	CM data items	F/N
1	**Devel. manager (DMgr)**	**Command Script**	**Generates target-specific set of commands to perform target builds**	**see lower-level steps**	**(66,125,124,134, 133,67,69,118,120, 121,135,160,162, 163,167)**	10
1.1	DMgr	Build instructions	Retrieves the "build instructions" CIs	na	relational database	

General instructions for building a CSCI are created and stored under CM. These instructions, which identify files and build parameters, may need to be tailored for specific platforms and locations of relevant files.

10 The Development Manager interacts with the Amdahl CM system, separate from the IVTE

Step	Operator	Element	Process Narrative	Int/f functions	CM data items	F/N
1.2	DMgr	Command scripts	Uses target-specific translators to convert generic build instructions to the command languages of the target platfoms	na	(Configuration ID of command scripts)	
1.3	DMgr	Build resources	Schedules the use of necessary IVTE resources	na	118,121,122,147, 148,151,150	11

These steps do not directly involve the IVTE; all interactions occur between the DMgr's workstation and the Amdahl. These steps are included to set the context for the actual data interactions.

1.4	DMgr	CI location data	Identifies the resources that are already resident in the IVTE	IVTE directory services	unique CI IDs, location data	
1.5	DMgr	Command script	Modifes the command script for this specific process	na	na	12

Step 2 Download and retrieve files

The DMgr activates the command script, causing the desired files to be retrieved and downloaded (if they are on the Amdahl) or retrieved in the IVTE.

2.0	**DMgr**	**Command script**	**Executes script on Amdahl and on IVTE**	**see lower-level steps**	**(135) 63,64**	13
2.1	Amdahl script processor	Command script	Establishes communications link with IVTE host	job-level support	transaction ID (job number)	14
2.2	Amdahl-SP	download script	Downloads command script for IVTE and files for script to operate on	File transfer	63,64	

11 Communications with the IVTE system manager are assumed to occur outside the scope of the Amdahl-IVTE interface, although a direct dialog connection is possible.

12 See comparable steps in the compile scenario.

13 See step 2 of the compile scenario

14 For purposes of handshaking and closure of CM actions, a transaction ID or job number is assigned to the process. This ID is used for confirmation, and to correctly link returned products with downloaded commands.

Step	Operator	Element	Process Narrative	Int/f functions	CM data items	F/N
2.3	IVTE script processor	file process script	Retrieves IVTE-based files (or points to them) for use in load-building	na	na	
2.4	IVTE-SP	IVTE-based files	Verifies that IVTE-resident files are identical to the controlled versions stored on the Amdahl	na	63,64,65	
2.5	IVTE host	**T-Driver** job	Finds file version mismatch, generates "failed job" status report	na	63,64,65	

The link transaction of this CSCI fails because the requested version of one of the required files was not available. The products therefore include the output which identifies the reason for failure, and the completion status that will close the CM operation.

The IVTE provides a response to the Amdahl that the operation is underway. There is no need for the communications link to remain open during the processing on the target platforms (although that may be the simplest approach). There is need for some form of acknowledgement of communications.

2.6	IVTE-SP	download status	Confirms successful receipt of all files and retrieval of local files	IVTE int/f Ack function	transaction ID (job number) 63,64,65	
2.7	Amdahl-SP	process status	Places transaction ID in "pending completion" status to await products	Amdahl int/f job mgr	(66,67,68,69,135, 148,153,154,155, 156,157,158,160, 162,163,166,169, 170,172)	

One side of the interface or the other, or both, need to track what jobs are in progress. This scenario assumes that both ends of the interface perform some transaction (job) management functions.

Step 3 Perform target builds

The script processor on the IVTE directs the appropriate jobstreams to the target platforms for **Monitor** and **T-Driver**. The former is downloaded to an IVTE workstation serving as a compile and link engine; the latter is processed on the IVTE host. Status and products are returned to the Amdahl upon completion.

Step	Operator	Element	Process Narrative	Int/f functions	CM data items	F/N
3.0	**IVTE**	**Monitor and T-Drive**	**Build target loads on specified platforms**	**see lower-level steps**	**na**	
3.1	IVTE script processor	**Monitor** job	Transmits command script and files to target platform	file distribution	66,67,68,69	
3.1.1	IVTE-SP	object files	Downloads object libraries and files as required	network int/f	66,67,68,69	
3.1.2	workstation (W/S) platform	Command script	Receives and processes command script	network int/f	na	15

The build environment for **Monitor** includes tools (e.g., linker), system libraries, custom shared libraries, and object files that are specific to this CSCI.

3.2	W/S platform	**Monitor** job	Builds executable from object files	na	na	
3.3	W/S platform	**Monitor** products	Uplinks product and status to IVTE host	network int/f	66,67,68,69,123, 124,125,126,132, 133,134,135,138, 147,149,150,151, 153,154,162,163, 173,174	

The linking of this CSCI is successful. Status therefore includes not only the fact of success but identification (name and version) of each system tool used in creating the product (e.g., linker, system library).

3.4	IVTE host	Products and status	Receives output and spools it for uplink processing	network int/f	na	16
3.5	IVTE-SP	products and status	Assembles job completion and CM records needed to complete the operations	na	na	

Step 4 Upload products

The IVTE host packages the products of the operations for uploading to the Amdahl. The transaction ID serves as a reference. CM-required information is formatted for use in creating new controlled files and annotating old ones.

15 This scenario assumes that any target-build platform will have the capability of receiving and executing command scripts via communications links.

16 The IVTE host acts as the interface between the IVTE and the Amdahl.

Step	Operator	Element	Process Narrative	Int/f functions	CM data items	F/N
4.0	**IVTE script processor**	**products and status**	**Package and upload products**	**see lower-level steps**	**63,64**	
4.1	IVTE-SP		Establishes or resumes communications session with Amdahl	IVTE int/f	na	
4.2	IVTE-SP	Job Status	Assembles and formats CM records to support CM operations on the Amdahl	CM data formatter	na	
4.3	IVTE-SP	Job products	Packages CM data, products, Job ID info	na	66,67,68,69,123, 124,125,126,132, 133,134,135,138, 147,149,150,151, 153,154,162,163, 173,174	
4.4	IVTE-SP	Product packages	Uploads packaged information to Amdahl	IVTE int/f	63,64	

Step 5 Process and store products

The Amdahl receives the products, directs them as appropriate, and submits information to the CM system to conclude the scenario.

5.0	**Amdahl-SP**	**Products**	**Performs recording and distribution**	**see lower-level steps**	**na**	
5.1	Amdahl-SP	Command script	Retrieves suspended script based on transaction ID returned with uploaded products	na	na	
5.2	Amdahl-SP	CM records	Processes CM records for submission to the CM system	CM formatter	na	17

17 The formatting tools on the Amdahl and the IVTE host are complementary; the functionality could be distributed to either system

Step	Operator	Element	Process Narrative	Int/f functions	CM data items	F/N
5.3	Amdahl-SP	CM products	Submits Monitor load as a derivative CI for CM	na	na	
5.4	Amdahl-SP	Process products	Distributes non-controlled products (e.g., listing files) as appropriate	na	na	

4.5 Scenario 3: Integration testing

Integration and qualification testing occurs in the IVTE using complete systems (CSCs or CSCIs) that interact with each other and with target hardware. The basic process involves building test environments (testbeds and hardware), using scripts and data to test the configuration, and reporting anomalies with Discrepancy Reports. The details of the process are reflected in the scenario below; they include location of test items prior to and during test, reporting to the configuration management (CM) system on the use of test items, reporting on status of tests, and promotion of tested components following successful qualification testing.

The SSE System Project is developing an extensive set of tools and capabilities to assist with planning, preparing, and analyzing the results of the testing process. (The test *execution* step is outside the scope of the SSE. Execution support tools will be developed elsewhere.) Those tools and capabilities are tightly integrated with the SSE-developed CM system that will reside on the GSDE Amdahl. While the SSE does not require any particular set of procedures for control and management of test articles, it does provide a system of attributes and relationships that are carefully matched to NASA Space Station Project procedural requirements.

The following scenario describes a relatively simple integration test. Three workstations are loaded with software and required to interact with each other and with external data sources and devices. The scenario is both simple and general to reflect conditions in either the SSTF or the SSCC. The scenario covers the process from assembly of test materials to recording the results. It includes creating testbeds, configuring the test environment, executing tests according to a test plan, reporting errors and successes, and collecting test results for later analysis and regression testing.

Figure 4 pictures this scenario.

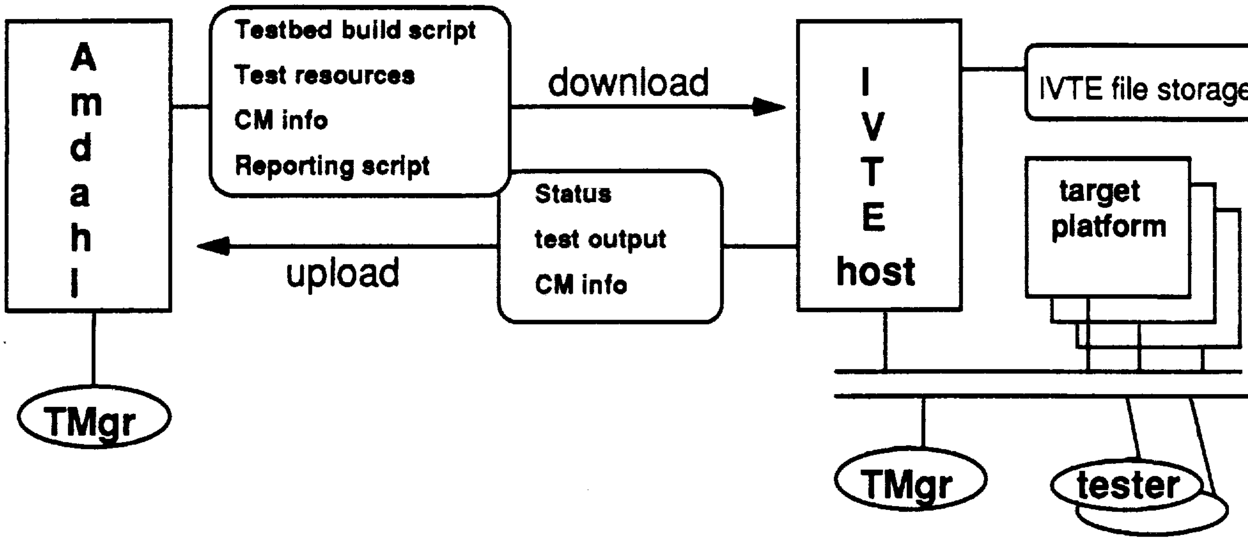

Figure 4. Integration testing scenario

Scenario context

The subject of the integration test is the software CSCI named **Monitor**. This workstation-resident CSCI provides the capability to read and control applications software in other workstations. To perform the test, two other CSCIs named **AppA** and **AppB** will execute according to test scripts, and appropriate data (e.g., a canned telemetry stream) will be provided. **AppA** and **AppB**, as well as the test data and test data drivers, have already been tested and are marked "passed test".

Step 1 Assemble test items

This scenario assumes that test setup definitions (e.g., test scripts, testbed parts lists, etc.) are created on the Amdahl and placed under CM. Development of these definitions logically precedes this scenario.

The person managing the test (TMgr, the test manager) uses tools on the Amdahl to specify and retrieve the parts list for the desired test sequence for CSCI **Monitor**. The parts list is used to generate a report on the status and location of all required test items. All items are marked "passed test" except the test article **Monitor** which is "ready for test". Several of the items are also marked "located in IVTE".

Step	Operator	Element	Process Narrative	Int/f functions	CM data items	F/N
1	**Test manager (TMgr)**	**Test item set**	**Assembles all necessary items to support a specific test session.**	**see lower-level steps**	**(66,67,69,117,118, 121,122,124,126, 135,138,147,173, 174)**	
1.1	TMgr	Test item list (report)	Generates a report on the CIs required for the specified test on CSCI **Monitor**	na	CI spec, test ID, item status	18
1.2	TMgr	Test item report	Determines that all items are known to the CM system and are marked "passed test"	na	CI test status (172)	19
1.3	TMgr	Test config. setup script	Identifies the resources that will be needed in the IVTE (e.g., target platforms)	na	specific CIs (66,67,69)	

These steps do not directly involve the IVTE; all interactions occur between the TMgr's workstation and the Amdahl. These steps are included to set the context for the actual data interactions.

Still using the Amdahl, the TMgr uses a tool to assemble all needed test items for downloading to the IVTE. The items are retrieved either physically (e.g., mounting a tape) or logically (e.g., providing a file specification). The IVTE-resident items, of course, are not physically moved. The test article (the CSCI to be tested) is checked out for testing. The test items (the supporting software, data, and tools) are flagged but not checked out; they can be used by others.

1.4	TM	Test Item set	Executes a procedure that either assembles, or verifies the accessibility of, all items in the test set		CI ID and location data	
1.4.1	TM	Test article (**Monitor**)	Checks out the executable file to be tested	na	CI test status	

18 The Test Manager interacts with the Amdahl CM system, separate from the IVTE

19 Using appropriate queries, the TMgr has followed a relational sequence from the test article (Monitor) to the "test status" fields of all of the CIs that are needed for testing.

Step	Operator	Element	Process Narrative	Int/f functions	CM data items	F/N
1.4.2	TM	Test items	Gets read-access to items which will support the test	na	Usage log	
1.4.3	TM	IVTE-resident test items	Verifies that the CIs residing in the IVTE are available, and are identical to the controlled files stored on the Amdahl	File system interface, IVTE file manager	(63,64,65)	

Question: will the Amdahl engage in a dialog with the IVTE host at this point to verify and reserve IVTE-resident items? Or will it simply note that certain items are to be retrieved during the test bed configuration process, and expect the IVTE host to abort if the resources are unavailable or have been modified?

Step	Operator	Element	Process Narrative	Int/f functions	CM data items	F/N
1.5	TMgr	Test resource allocation	Identifies needed physical (not software) resources and schedules them for the test	na (see note)	(166)	20
1.6	TMgr	Operator Test Scripts	Generates any necessary hardcopy for test setup and execution	na	na	
1.7	TMgr	Target platform Build scripts	Creates target-specific download script to perform file transfers and command operations	target-based script translator	use of V&V tools	

Step 2 Establish test environment

The test manager (TMgr), interacting with the Amdahl, uses the testbed build instructions to create the testbeds and test environment in the IVTE. In SSE terminology, a testbed is an integrated set of software including the software which is the subject of the test. The test environment consists of software and hardware.

20 It is assumed that physical resource allocation is NOT performed electronically through the Amdahl-to-IVTE link. If some sort of automatic scheduling and allocation of resources is intended, the Amdahl-IVTE interface needs to reflect that fact.

Step	Operator	Element	Process Narrative	Int/f functions	CM data items	F/N
2.0	**Test manager**	**Testbed, test environment**	**Executes command scripts to perform necessary configuration and allocation of resources**	**see lower-level steps**	**na**	
2.1	TMgr	Download command script	Modifies the file transfer script to specify particular resources in the IVTE (e.g., the actual target platforms)	na	Resource IDs	21
2.2	TMgr	Modified command script	Saves the modified script for future use and for test replication	na	Test event log CM assigns Test Configuration ID	

The test item set includes testbed build instructions ("command scripts") to direct the file transfer, target loading, data transfer, and initiation of testbed software. Once constructed, this command script would be saved for repeat use until a phase of testing is complete. This scenario assumes that this script would be placed under CM, but the issue is not central to this scenario.

2.3	TMgr	Download command script	Executes the script on the Amdahl (and indirectly on the IVTE host) to effect the downloading and configuration of test resources.		na	
2.3.1	Amdahl script processor	Comm link	Establishes a dialog with the IVTE host to download scripts and files	Comm int/f	na	
2.3.2	Amdahl-SP	Test Items	Downloads all Amdahl-based test resources to IVTE	file transfer	(63,64,65)	
2.3.3	IVTE script processor	command script	Distributes Test Items (scripts and files) to appropriate targets	cmd interpreter and Ack	na	
2.3.4	IVTE-SP	command script	Retrieves IVTE-based Test Items and processes them	Ack	na	

The TMgr executes the command lists to download the test items and configure the test environment. No errors are reported.

Unlike the compile and build scenarios which were essentially batch-type jobs, this scenario requires confirmation and verification at many steps of the process. A resource which is needed late in the test must usually be allocated and verified early.

21 This step may be automatic, drawing on the scheduling system for the IVTE and the file locator function of the CM system. This scenario assumes that this step would be manual.

Question: what acknowledgment procedures will exist to verify that procedures, e.g., downloading an executable file into a workstation, have succeeded? Will there just be a manual verification, or will there be a handshaking protocol?

> Note: based on our analyses and the information previously provided, it is assumed that all communications between the Amdahl and the IVTE occur via the IVTE host. A full acknowledgement protocol requires that the IVTE host execute its portion of a command list, get confirmation from the target computer, and then reflect that confirmation back to the Amdahl.

Step	Operator	Element	Process Narrative	Int/f functions	CM data items	F/N
2.4	IVTE script processor	Execution command scripts	Sends execution scripts to loaded software to perform specified testing	network interface	na	22
2.6	IVTE-SP	Operator command scripts	Provides instructions to the operators who will interact with Monitor, AppA, and AppB	na	(134)	

The set of test items also includes execution scripts and testing instructions for all participants in the test sequence. The test scripts are downloaded to the appropriate drivers after confirmation that the drivers are activated (see question, above). The test instructions are delivered to the test team.

Question: will testers get their instructions on paper? On portable computers? On development workstations colocated with the IVTE target workstations? Will the test instructions be downloaded to the IVTE or will they stay in the Amdahl domain?

Alternate outcome: It may be that the attempt to construct the testbed fails for some reason. For example, one of the required IVTE-resident software loads may not pass a CRC check (i.e., has been modified in the IVTE); or perhaps some necessary element was left out of the test environment definition (TED). In this case, the IVTE will report back to the Amdahl to change the status of some element(s). Possibly a DR would also be generated to fix an omission in the TED.

The IVTE would need to report back in some fashion (e.g., automatically, manually, by telephone...) so that the status of test items can be updated in the CM database.

22 The startup process may be automatic or manual; the IVTE side of the interface needs to support either capability unless the scripts themselves include it.

Step 3 Reporting on test execution.

The testing session involves following a sequence of test activities to demonstrate that **Monitor** performs as required. The typical integration and qualification processes will involve several computers, test drivers, operators, and data elements. The operation is controlled (in this scenario) by a Test Director (TD) in the IVTE area who directs and monitors the performance of the testing.

> Please note that our concern with the test execution phase lies with the *reporting* of results back to the CM system, and not with the test execution itself. Unless the testing sequence is to be controlled interactively from workstations on the Amdahl (via the Amdahl to IVTE link), test execution is outside the scope of this analysis. It is described for the sake of context and continuity.

The TD fires up the monitored applications, **AppA** and **AppB**, along with the telemetry stream and the device interface mechanism. Following the test commmand scripts (automatic execution) and test operator scripts (manual intervention) the test team proceeds through the test sequence.

Step	Operator	Element	Process Narrative	Int/f functions	CM data items	F/N
3.0	**Test Director**	**Test definition**	**Initiates, monitors, and reports on test execution**	**see lower-level steps**	**na**	
3.1	TD	Command lists	Starts up all test resources per test build script	na	118,120,122,124, 125,126,132,133, 134,138	
3.2	Test personnel	Test scripts	Step through test sequence	na	na	
3.3	Test personnel	Test envi-ronment	Modifies test environment to enable test conditions to be met	na	127,128,129,130, 131	
3.3.1	TD	Test item	Generates a report on changes needed	na	CI ID	

The test environment definition and test scripts may not provide adequate detail, or may not establish conditions needed to test all capabilities. In such instances, test items may be modified during a test to achieve the desired results.

A hypothetical example: The **Monitor** program must be able to read and clear a "data overrun" alert in **AppA**. Because **AppA** is so much more efficient than expected, the specified test data stream is unable to produce the alert. The test team changes the driver settings to send the data faster.

In practice, there will often be instances where the test item is modified in the IVTE for the purpose of gaining as much information as possible about a discrepancy. Such modification may also be performed to permit informal testing of the test article to proceed. This process may lead to the test-item failure noted as a possible outcome of step 3.4.

The testing process is designed to produce pass/fail indications for each step in the sequence. These indications may not be available until output data has been analyzed, but sooner or later test reports are due to close out the test session.

Step	Operator	Element	Process Narrative	Int/f functions	CM data items	F/N
3.3	TD	Test report	Reports on outcome of each test in the sequence	CM reporting	CI ID, test ID, test level, status 167,168,169,170, 172,135	

Reporting may occur in any (or all) of several different ways. The IVTE host may provide an online reporting mechanism. The reports may be recorded on paper and entered manually on the IVTE (or an IVTE w/s) and uploaded to the Amdahl. Or the reports could be transmitted in hardcopy to the IVTE and entered using the standard CM tools.

This scenario assumes that the SSE capabilities described for OI 6.0 will be utilized. This includes traceability to test cases, test conditions, configuration items, and change instruments. The reporting process is presumed to operate through the IVTE- Amdahl interface.

Question: will CM status reports be transmitted via the Amdahl-IVTE link? If so, via what mechanism and in what format?

Note that the test status reports involve a large amount of information (such as the IDs of many CIs) that is probably available online in the IVTE host. A forms-filling approach to reporting would seem feasible.

Alternative A: CSCI **Monitor** passes all of its tests. When the test sequence is finally complete and successful, the TD generates a report that each step was passed.

3.4A .1	TD	Test report	Reports that all items in sequence passed	CM data formatter	167,168,169,170, 172,135	
3.4A .2	TD	CM status	If this completes the test plan for this CSCI, the test status is updated in the Amdahl CM system	CM reporting	na	

In order to support regression testing, the entire configuration must be recorded (though not necessary stored), including any on-the-fly changes made to the test configuration. The status of **Monitor** is changed to "passed test".

Alternative B: CSCI **Monitor** passes some of the tests. The testing process is not complete, but intermediate results are reported. Some tests may have been skipped. Tests which failed will generally lead to Discrepancy Reports.

Step	Operator	Element	Process Narrative	Int/f functions	CM data items	F/N
3.4B .1	TD	Test Report	Reports on all subtests which were passed.	CM formatter	167,168,169,170, 172,135,154,158, 165	
3.4B .2	TD	Discrepancy Report	Initiates DR for failing tests, including whatever helpful information can be acquired through testing	na	DR or Non-Conformance Report	23

A possible outcome is that one of the test configuration items (e.g., **AppA**, or the canned telemetry stream) exhibits an anomaly or causes a test to fail. In this case, the status of the test item (maintained in the Amdahl CM system) must be changed to reflect the situation. A DR may also be required to correct the situation.

Alternative C: There is a problem of such severity that the test is cancelled without conclusion. (For example, a failure of one of the test items or hardware elements might lead to cancelling the test session). All test articles which were marked as "in use" are released; any identifiable problems are reported.

Step	Operator	Element	Process Narrative	Int/f functions	CM data items	F/N
3.4C .1	TD	Test Report	Cancels the test without formal reporting, restores test status of **Monitor**	CM reporting	172,167,135	
3.4C .2	CM system	Test status fields	Rolls back all "in use" indicators to previous states (as if testing had not occurred)	CM reporting	172,167,135	
3.4C .3	TD	DR	Files report on testing problems	na	DR or Non-Conformance Report	

It may be that a simple "cancel the test" message can be sent to the Amdahl CM system to effect this set of operations. Somewhere that message would be expanded into a full set of "Release this resource" database commands.

23 The DR process is presumed to be independent of the Amdahl-IVTE interface.

Step 4 Test data capture.

In cases where a test sequence produces archivable output, it may be necessary to capture such output and move it to the Amdahl for controlled storage. The output may be used in regression testing, or it may be used for analysis of the system being tested. In either case, there will be a mechanism for creating the appropriate CM data record describing the output, and uploading the file and the CM data to the Amdahl.

Note: no assumption is made about where the output analysis tools reside. The configuration management function, however, needs to get data produced in the IVTE (such as tool identification) to ensure that later tests and analyses are consistent.

Step	Operator	Element	Process Narrative	Int/f functions	CM data items	F/N
4.0	**TD**	**Test Output**	**Archives test output and places it in controlled storage on the Amdahl**	**see lower-level steps**	**na**	
4.1	TD	Test output	Defines CM records for each configuration item to be controlled	CM formatter	na	
4.2	TD	CIs	Uploads files and CM records to Amdahl	File transfer	66,67,68,69	

The CM records include information on the tools used, relationships to tests, user ID, related CSCI, etc. The information describing these files serves as a packing slip for the CM system. Since most of the information will already exist in the IVTE, this scenario assumes that the files and descriptors are uploaded together.

4.6 Compendium of issues and questions

The assumptions, findings, and questions presented in the three scenarios are assembled here for convenient reference. The assumptions and questions are numbered to facilitate responses by reviewers. Page numbers refer to discussion of questions or assumptions in the scenarios.

4.6.1 Assumptions

1) A mainframe in the IVTE will act as an IVTE host computer, coordinating the interface with the Amdahl and buffering file transfers. Additional functionality is TBD (figure 1, page 6).

2) The CM-access tools (such as dynamic and static analysis tools) provided by the SSE will be used in the SPEs (page 13).

3) IVTE platforms will be capable of hosting software that supports a "remote batch" processing capability (note, page 17).

4) Code development through the unit-test stage (at least) occurs in the SPEs (page 12).

5) IVTE platforms will be used for some compilation and target load building (page 12).

6) Some of the products of the compile/link/test process will be retained on, or returned to, controlled storage in the IVTE to minimize downloading from the Amdahl (page 8).

7) Duplicated storage in the IVTE will be tracked in, or managed from, the Amdahl (page 8).

8) A processing transaction over the Amdahl-IVTE interface will be assigned some unique identifier that can be used on both ends of the transaction (for example, a job number or job name) (page 16).

9) File verification mechanisms provided by the SSE (e.g., checksum and CRC determination, file comparison) will be available in both the Amdahl and the IVTEs (page 16).

10) Testbed definitions (list of software configuration items and instructions for assembling them for testing) are controlled in the Amdahl (page 29).

11) All software and data items that materially contribute to a test session are held in controlled storage on the Amdahl, with appropriate records in the CM database. *Copies* of these items may also be kept in the IVTE, as noted above (page 29).

12) There is some mechanism for recording that a configuration item is "in use" in a testbed (page 29).

13) Scheduling of IVTE resources (e.g., a workstation platform needed for integration testing) is *not* part of the Amdahl-IVTE interface (note, page 30).

14) The discrepancy reporting (DR) process is not part of the Amdahl-IVTE interface (note, page 34).

15) No assumption is made about the user's dialog with the Amdahl. For testing, the user is probably colocated with the IVTE, but uses a workstation that is networked to the Amdahl and not to the IVTE (page 28).

16) The actual execution of a testing session can occur with the IVTE logically disconnected from the Amdahl. Status reporting and data collection can be

performed in a standalone mode, and communicated to the Amdahl after a test session is completed (page 32).

4.6.2 Interface data flows

The major data flows identified in these scenarios are:

- configuration items (source code, object files, test articles, test scripts, etc.) from Amdahl to IVTE
- configuration data (concerning CIs) from Amdahl to IVTE
- file transfer (and perhaps directory information) from IVTE to Amdahl
- status of processing from IVTE to Amdahl
- process outcome reporting from IVTE to Amdahl
- test output and new/derived configuration items from IVTE to Amdahl
- configuration descriptive information (for CI products) from IVTE to Amdahl

4.6.3 Interface functionality

Operational interface requirements implied by these scenarios include:

- bidirectional file transfer (and conversion, if needed by different format computers) with confirmation and directory inquiry capabilities
- capability for the Amdahl to generate, and the IVTE host to execute, command scripts to direct file operations, binary loading, and execution of processes on computers in the IVTE
- IVTE host ability to format CM records and reports (eg, change status of item **AppB** to "not ready for test")
- co-operative Amdahl capability to process data into the CM system
- capability to record and recall session data (eg, the system-unique CM identifier of a test item) for the purpose of completing and annotating reports; either Amdahl or IVTE host or both, depending on design

4.6.4 Interface questions

The questions raised in these scenarios are listed below.

1) Processing transactions (Amdahl-->IVTE-->Amdahl) will require some form of command language. What will this language (or these languages) consist of? Will it be the User Interface Language to be used (when defined by the SSE)? Will it be the command language(s) of the native OSs of IVTE platforms (e.g., IBM JCL, DEC VAX DCL)? Will there be one or several such languages? (page 14)

2) Are the requirements for the IVTE platform command language(s) already specified or should we make assumptions and recommendations thereto? (page 14)

3) In the same vein, will command procedures be packaged and transmitted (a black box approach) and then processed on the receiving end? Or will there be a transaction dialog to support interactive operations? (page 16)

4) How will users identify the files and versions of controlled items which are duplicated in the IVTE? (Is it a manual process or will there be automatic directory mechanisms?) (page 14)

5) What level of control will be exercised over files and command scripts that are to be downloaded to the IVTE? Will such files need to be placed under CM before they can be downloaded? (page 15)

6) Typical jobstream scripts need to be tailored for specific circumstances such as specifying which compile engine to use, locating shared files, and including user identification and authorizations. Can this tailoring of controlled files occur *without* formal CM (i.e., retrieve the command script, modify it, execute it), or will the modified scripts be subject to CM? (page 15)

7) What mechanism will be used for tailoring command scripts (e.g., text editors, forms-filling tools, parameter lists)? (page 15)

8) What mechanism(s) will be used to assign and track transaction identifiers? Will this occur in the Amdahl software, IVTE software, or manually by the user? (page 17)

9) Where in the overall transaction process

Amdahl-->IVTE host-->platform-->IVTE host-->Amdahl

is configuration identification data generated? Where is it formatted for entry to the CM system? (page 18)

10) Which computers involved in the interface have the capability to initiate a transaction (the Amdahl, the IVTE hosts, IVTE platforms)? Can this initiation be automatic or is manual intervention required? (page 19)

11) During creation and operation of a testbed, where some of the required items are expected to be found in the IVTE: can the Amdahl perform some advance reservation and verification to ensure that all items are available and intact? (Halfway through a two-hour test is the wrong time to discover that a needed dataset is not available!) (page 29)

12) What sort of acknowledgement mechanism will exist to confirm that operations within the IVTE (e.g., loading software onto a platform) were successfully completed? (Subsequent operations in a script may make no sense and waste resources if prerequisite operations are not successful.) (page 31)

13) How will scripts and instructions be provided to testers? Will this information be transferred over the Amdahl-IVTE link? (page 31)

14) What mechanism will be used to return test-status reports to the Amdahl in order to update the "test status" fields of controlled items? (page 33)

15) During assembly of resources, will the Amdahl engage in a dialog with the IVTE host to verify and reserve IVTE-resident items? Or will it simply note that certain items are to be retrieved during the test bed configuration process, and expect the IVTE host to abort if the resources are unavailable or have been modified? (page 14)

5.0 CM information fields

The table on the following pages identifies the information (attributes, or "fields") that the SSE-provided CM system can record and track about a configuration item. The numbers in the left column are keyed to the **CM Information** column in the scenarios. There are many numbers which are skipped (such as the first 62) because they did not seem to be information that would be communicated over the Amdahl-IVTE interface.

The second column provides a description of the information element. The third column contains our best guess as to how the information will generally be supplied to the Amdahl CM-system. The fourth column will eventually contain a reference to the user or class of user who is responsible for providing the attribute values.

	CM Field Title	Field Input Auto/Manual	Responsible Party
63	File Compare	Auto	
64	CRC Field Comparison	Auto	
65	Checksum Comparison	Auto	
66	Configuration Item Identifier	Auto	
67	Configuration Item Name	Auto	
68	Configuration Item Description	Manual	
69	Configuration Item Version	Auto	
117	Software Integration Hierarchies	Auto	
118	Testbed Software Configuration	Manual	
120	Type of Transaction Performed	Manual	
121	Configuration ID Information of the Test Resource Affected	Auto	
122	Status of Test Resource Before Transaction	Auto	
123	Status of Test Resource After Transaction	Auto	
124	USERID of Person Performing Transaction	Auto	
125	Date and Time Of Transaction	Auto	
126	Config. ID Info. Of the Config. Item Affected	Auto	
127	Status of Config. Item Before Modification	Auto	
128	Status of Config. Item After Modification	Auto	

	CM Field Title	Field Input Auto/Manual	Responsible Party
129	USERID of the Person Performing the Modification	Auto	
130	Date and Time of Modification	Auto	
131	Reason For Modification	Manual	
132	Test Tools	Auto	
133	Test Data	Auto	
134	Test Scripts	Auto	
135	Config. ID Info. Of the Test Invoked	Auto	
136	Functional Requirements Implemented	Manual	
137	Name of Analyst	Manual	
138	Test Resource Class	Manual	
139	Resource Class/Relationship Value	Manual	
140	Resource Class/Attribute Value	Manual	
147	Config. ID of Config. Items Being Tested	Auto	
148	Status of all Config. Items Before Testbed Build	Auto	
149	Status of all Config. Items After Testbed Build	Manual	
150	USERID of Person Building Testbed	Manual	
151	Date and Time of Testbed Build	Auto	
152	For Test Results: Type Of Transaction Performed	Manual	

	CM Field Title	Field Input Auto/Manual	Responsible Party
153	Config. ID Info of the Test Whose Results are being Posted	Auto	
154	Config. ID Info of the Config. Items Tested		Auto
155	Status of all Affected Config Items Before Posting	Auto	
156	Status of all Affected Config Items After Posting	Manual	
157	USERID of person Posting Results	Manual	
158	Date and Time Results Posted	Auto	
159	Optional Remarks	Manual	
160	USERID of person Executing Test Procedure	Manual	
161	USERID of person Authorizing bypass of the Previous test in a test Sequence	Manual	
162	Date of Test Execution	Auto	
163	Time of Test Execution	Auto	
164	Config. Items ID of Products Tested	Auto	
166	Current Status of Test Resources: Under Development, In Test, Ready of Test, Completed Test, Ready for Test with Bypass, Failed Test	Manual	
167	Test Procedure Identifier	Auto	
168	Component placed in Test	Auto	

	CM Field Title	Field Input Auto/Manual	Responsible Party
169	Period the Components Were Under Test	Auto	
170	Number of Test Failures For Each Component	Manual	
172	Test Results once the Testbed is Successfully Built Passed Test, Failed for Rework, Failed for Retest, Failed with Bypass, Test Bypassed, Defective Test	Manual	
173	Configuration Identification Sensitivity Levels: 0-Negligible impact; 1-Minimal Impact; 2-Adverse Impact; 3- Irreparable Impact	Manual	
174	Security Information 1-Personal 2-Financial, Commercial, Trade Secret 3- NASA Internal Operations 4- Investigation, Intelligence Related, Security 5- Other Federal Agency 6- Unclassified National Security-Related 7- National Resource Systems 8- Mission Critical 9- Operational 10- Life Critical 11- High or New Technology 12- Other Unclassified	Manual	

Appendix A - CM reporting

Title of CM Report	Title of Field/Information
Configuration Item Version Description Report	
	Configuration Item Identifier
	Resolved Change Instruments
	Unresolved ChangeInstruments
	Deviation Numbers
	Wavier Numbers
	Version Number
	Version Description
	Status
Action Item Reports	
	Status Date
	Action Item Number
	Action Item Description
	Assignee
	Required Completion Date
	Meeting Number
	Meeting Date
	Meeting Type
CR/DR Number	
	Closure Date
	Reason for Closure

Title of CM Report	Title of Field/Information
Deviation/Waiver Report	
	Deviation or Waiver Number
	Title
	Description
	Associated Configuration Item Identifiers
	Approval Authority
	Approval Date
	Originator Organization
	Proposed Effectivity
	Related CR/DR Numbers
	Related Deviation/Waiver Numbers
	Requirements Affected
	Applicable Documents
	Rationale/Risk
	Assessment
	Explanation
	Schedule Effects
	Cost Reduction/Increase Estimates
Release by Platform Report	
	Platform
	Release
	Open CR/DR Numbers
	Incorporated CR/DR Numbers
	CR/DR Titles
	CR/DR Descriptions

Title of CM Report	Title of Field/Information
Action Item Meeting Report	
	Action Item TItle
	Action Item Number
	Action Item Version Number
	Review Meeting Number
	Review Meeting Date
	Review Meeting Type
	Action Item Description
	Assignee
	Required Completion Date
	Action Item Status
	Status Date
	CR/DR Number
	Closure Date
	Reason for Closure
Deviation or Wavier Report	
	Deviation or Waiver Number
	Title
	Description
	Configuration Item Identifier
	Approval Authority
	Approval Date
	Originator Organization
	Originator Name
	Proposed Effectivity
	Related CR/DR Numbers
	Related Deviation/Waiver Numbers
	Requirements Affected
	Applicable Documents

Title of CM Report	Title of Field/Information
	Rationale/Risk Assessment
	Explanation
	Schedule Effects
	Cost Reduction/Increase Estimates
Change Requests	
	Unique Identifier
	Revision Number
	CR Number
	CR Title
	Initiator
	Initiator Center/Office
	Initiator Phone Number
	Priority of the CR
	Status of the CR
	Submitter
	Submitter Center/Office
	Submitter Phone Number
	Affected Document(s) Number(s)
	Affected Document(s) sub ID(s)
	Affected Document(s) Revision(s)
	Affected Document(s) Title(s)
	Reference Documentation
	Description Change
	Reason for Change
	CR Assignee
	Date CR Assigned
	Assignee Phone Number
	Program Area(s) Affected
	Change Impact

Title of CM Report	Title of Field/Information
	Cost Impact
	Work Breakdown Structure
	Effectivity (the release when the change should be implemented)
	Impact Description
	Recommendation/Remarks
	Board Member Authorization For CR Submission
	Date of Submission Approval
(Information on CR Implementation Instructions)	
	Recommended Implementation for CR
	Board Level
	Contractor CMO Receipt Date
	Next Scheduled Activity
	Transmittal Required (Y/N)
	Outgoing Transmittal Number
	Incoming Transmittal Number
	Implemented (Y/N)
	Date CR is Implemented in a Release
	Date Approval is Required
	Analysis Due Date
	Related Change Instruments
	Board Disposition of Recommended Implementation
	Board Meeting Number
	Board Meeting Date
	Board Chairman Signature
	Board Disposition Date

Title of CM Report	Title of Field/Information
(Information on CR Analysis)	
	CR Analysis Engineer
	Description of Change
	Workarounds or Recovery Procedures
	Requirements Associated with CR
	Alternative Solutions
	Issues
	Impact if not Implemented
	Recommendation Based on Analysis
	Estimated Schedule Impact
	Estimated Lines of Code
	Estimated Total Cost
	Funding Available in the Current Budget
	Fiscal Year of Available Funding if Yes
	Recommended Release Number
	Procurement Activity Required (Y/N)
	Shipping Instruction Required (Y/N)
	Transmittal Required (Y/N)
	Dependencies
(Information on CR Cost Analysis)	
	Estimated Initial Material Cost
	Estimated Material Maintenance Cost
	Estimated Labor in Manweeks
	Comments

Title of CM Report	Title of Field/Information
Discrepancy Reports (General Information)	
	Unique Identifier
	Revision Number
	DR Number
	DR Title
	Date Written
	Category of DR
	Priority (Recommended)
	Status of DR
	Originator Name
	Originator Organization
	Originator Phone Number
	Originator USERID
	Date/Time at which Problem Occurred
	Location/SPF at which Problem Occurred
	Nonconforming Platform (SW/HW) Name
	Nonconforming Platform (SW/HW) Version
	Description of Problem
	DR Assignee
	Date DR Assigned
	Assignee Phone Number
	Results of Investigation/Recommended Resolution
	Board Disposition of DR
	Board Meeting Number
	Board Meeting Date
	Board Chairman Signature
	Board Disposition Date
	Release Problem Found On
	Release Problem Introduced On

Title of CM Report	**Title of Field/Information**
(Information on DR Implementation Instructions)	
	Recommended Release for DR Implementation
	Affected Document(s) Revision(s)
	Affected Document(s) Title(s)
	Comments
	Board Level
	CMO Receipt Date
	Next Scheduled Activity
	Transmittal Required (Y/N)
	Outgoing Transmittal Number
	Incoming Transmittal Number
	Implemented (Y/N)
	Date Implemented
	Date Approval Is Required
	Analysis Due Date
	Assessment Type
	Related Change Instruments
	Test Procedure Number In Which this DR was Detected
	Life Cycle Phase in which This DR was Detected
	Test Step Number in which This DR was Detected
	Retest Required (Y/N)
	Level of Test if Yes
	SR&QA Recommended Disposition
	SR&QA Disposition Signature
	SR&QA Date
	SR&QA Rationale
	Reason for DR Closure
	SR&QA Closure Signature
	Date of SR&QA Approval

Title of CM Report	**Title of Field/Information**
(Information on DR Analysis)	
	DR Analysis Engineer
	Description of the Problem
	Cause of the Problem
	Workarounds or Recovery Procedures
	How does the User See Effect
	Scenario to Produce Problem and its Probability
	Problem Solution
	Corrective Action for Cause Of Problem
	Impact if Not Implemented
	Recommendation Based on DR Analysis
	Estimated Schedule Impact
	Estimated Lines of Code
	Estimated Total Cost
	Life Cycle Phase Where Discrepnacy was Introduced
	Recommended Release Number
	Procurement Activity Required (Y/N)
	Shipping Instructions Required (Y/N)
	Transmittal Required (Y/N)
	Dependencies
(Information on DR Cost Analysis)	
	Estimated Initial Material Cost
	Estimated Material Maintenance Cost
	Estimated Labor in Manweeks
	Comments
Traceability Report	
	Configuration Item Identifier

Title of CM Report	Title of Field/Information
	Traceability Relationship Identifier
	Traceability Object Identifier
	Traceability Relationship Type
Requirements Traceabilty Report	
	Traceability Object Identifier
	Traceability Object Version Number
	Traceability Relationship Identifier
	Traceability Relationship Type
Traceability Hierarcy Report	
	Parent Configuration Item Identifier
	Traceability Relationship Identifier
	Traceability Relationship Type
	Children Configuration Item Identifier
	Parent Configuration Item Identifer
	Number of Levels
Sibling Traceability Report	
	Sibling Traceability Object Identifiers
	Traceability Relationship Identifiers
	Traceability Relationship Types
	Child Traceability Object Identifier
Test and Requirements Traceability Report	
	System Test Identifier
	Requirements Identifier
	Requirement Name

Title of CM Report	Title of Field/Information
Configuration Items Affected by A Non-Conformance Report	
	Non-Conformance Report Identifier
	Description
	Configuration Item Identifiers
Non-Conformance Report	
	Title
	Date and TIme of Non-Conformance
	Location (Facility/Site)
	Non-Conformance Report Identifier
	Test or Operation Being Performed At Occurrence
	Prevalent Conditions
	Non-Conforming Item Identifier
	Associated configuration ItemIdentifier
	Contractor Deliverable End Item
	Symptoms of Non-Conformance
	Description of Non-Conformance
	Criticality of Non-Conformance
	Non-Conformance Category
	Cause of Non-Conformance
	Test/procedure Identifier Used During Non-Conformance
	Subsystem and Other Affected Modules
	Indication if Non-Conformance is A Failure or Unsatisfactory Condition
	Originator Data
	Indication is a Generic Trend Has Been Established
	Next Higher Assembly Identifier
	All Items that May Be Affected By The Non-Conformance
	Planned Date of Resolution

Title of CM Report	Title of Field/Information
	CSCI Affected
	CSC Affected
	Life Cycle Phase in Which Non-Conformance was Detected
	Individual and Organization Assigned to Resolve Non-Conformance
	Priority of Non-Conformance
	Indication If Retest Required
	Contract Number
	Status of All Data
	Date of Last Update
	Status of Progress in Identifying and Correcting Non-Conformance
	Constraint Requiring Resolution
	NASA SRM&QA Status and Comment
	Status of Resolution (Open/Pending)
Non-Conformance Closeout/ Explanation Report	
	Title
	Date and TIme of Non-Conformance
	Location (Facility/Site)
	Non-Conformance Report Identifier
	Test or Operation Being Performed At Occurrence
	Prevalent Conditions
	Non-Conforming Item Identifier
	Associated configuration Item Identifier
	Contractor Deliverable End Item
	Symptoms of Non-Conformance
	Description of Non-Conformance

Title of CM Report	Title of Field/Information
	Criticality of Non-Conformance
	Non-Conformance Category
	Cause of Non-Conformance
	Test/procedure Identifier Used During Non-Conformance
	Subsystem and Other Affected Modules
	Indication if Non-Conformance is A Failure or Unsatisfactory Condition
	Originator Data
	Indication is a Generic Trend Has Been Established
	Next Higher Assembly Identifier
	All Items that May Be Affected By The Non-Conformance
	Planned Date of Resolution
	CSCI Affected
	CSC Affected
	Life Cycle Phase in Which Non-Conformance was Detected
	Individual and Organization Assigned to Resolve Non-Conformance
	Priority of Non-Conformance
	Indication If Retest Required
	Contract Number
	Release Document Responsible For Implementing Corrective Action
	Results of Analysis and Tes In Isolating and Diagnosing Non-Conformance
	Remedial and Corrective Action
	Time/Cycles in use if Applicable
	Date of Resolution
	Related Non-Conformance Identifiers
	Explanation Rationale
	Efforts made to Determine Non-Conformance Cause

Title of CM Report	Title of Field/Information
	Assurance that Explanations do not Negate Each Other
	When Last Test of Item is to be Performed
	Effect on SSFP and SSE System If Non-Conformance Recurred And Recommended Work-Around
	Effectivity of Non-Conformance Closeout/Explanation
	Level of Retest Required
	SMR&QA Closure Approval
	NASA Assignee
	NASA SRM&QA Approval
	Status of Resolution (Closed/Explained)
Critical Items List	
	Configuration Item Identifier
	Required Fault Tolerance
	Sum of the Implemented Fault Tolerance of the Children Configuration Items
Expanded Critical Items List	
	Configuration Item Identifier
	Configuration Item Description
	Required Fault Tolerance
	Implemented Fault Tolerance
	Failure Mode
	Failure Effects
	Implementation Rationale
	Sum of the implemented fault Tolerance of the Children Config. Items

Title of CM Report	Title of Field/Information
Common Cause Report	
	Configuration Item Identifier
	Required Fault Tolerance
	Implemented Fault Tolerance
	Best Guess Probability of Failure
	Proven Probability of Failure
	Parents' Configuration Item Identifier
	Parents' Required Fault Tolerance
	Parents' Implemented Fault Tolerance
	Parents' Best Guess Probability Of Failure
	Parents' Proven Probability of Failure
	Configuration Items with a Relationship with more than one Parent
IT&V Current Test Resources Report	
	Resource Class
	Resource Class/Relationship Value
	Resource Class/Attribute Value
	Attribute Values
	Relationship
	Attribute/Relationship Values
	Attribute
	Relationship
IT&V Test Status Report	
	Test Procedure Used
	When Configuration Item was Last Modified
	Level of Test Procedure (component, integration, system)
	When Test Resource was Last Modified

Title of CM Report	Title of Field/Information
Testbed Build Report	
	Type of Transaction Performed
	Configuration Identification Information of the Test Invoked
	Configuration Identification Information of the Configuration Items Being Tested
	Status of all Affected Configuration Items Before the Testbed Build
	Status of all Affected Configuration Items After the Testbed Build
	USERID of the Person Building the Testbed
	Date and Time of the Testbed Build
Test Results Report	
	Type of Transaction Performed
	Configuration Identification Information of the Test whose results Are being Posted
	Configuration Identification Information of the Configuration Items Being Tested
	Status of all Affected Configuration Items Before Posting
	Status of all Affected Configuration Items After Posting
	USERID of the Person Posting The Results
	Date and Time of the Results Posting
Test Resource Status Report	
	Configuration Identification Information of the Test Resource
	Under Development
	In Test
	Ready for Test

Title of CM Report	Title of Field/Information
Testing Status of Development Components Report	
	Configuration Identification Information of the Component
	Under Development
	In Test
	Ready for Test
	Ready for Test with Bypass
	Completed Test
	Failed Test
Test History for a Component Report	
	USERID of the Person Performing The Test
	Date of the Test
	Time of Test Execution
	Test Procedure Identifier
	Component Place in Test
Test Metric Report	
	Component Identification
	Period the Component was under Test
	Number of Test Failures for Each Component
	Components with Above Average Failure Rates

Appendix B - CM tools

SSE Provided Tools	Tool Information/Capability
Quality Assurance Media Tools	
	Compare a File or Group of Files With a Copy of that File or Group Of Files
	Compare CRC/Checksum Field Between Files or Groups of FilesWith Copy of Files or Group of Files
	Message of Comparison Between Two Files, Two CRC Fields, or To Checksum Fields
Software Fault Tolerance Analysis Tool	
	Perform Fault Tolerance Analysis Enter, Store, Modify, Retrieve,or
	Delete Fault Tolerance Information That Follows:
	Configuration Item Identifier
	Configuration Item Name
	Configuration Item Description
	Configuration Item Version
	Required Fault Tolerance
	Implemented Fault Tolerance
	Failure Mode
	Failure Effects

SSE Provided Tools	Tool Information/Capability
	Functional Requirements Implemented
	Name of Analyst
	Implementation Rationale
	Analyst Assumptions
	Mechanism Used to Provide Fault Tolerance
	Possible Causes of Fault Safeguards, and Test for Prevention
	Mechanism for Isolation and Recovery
	Retention Rationale
	Summary of Maintenance Procedure For Replacement and Verification
	Degree of or Effect on Fault Tolerance During Replacement and Verification
	Deviation/Wavier Number
	Best Guess Probability of Failure (Decimal Fraction)
	Proven Probability of Failure (Decimal Fraction)
	Redundancy Flag
	Parent Relationships (Configuration Item Identifier)
	Child Relationships (Configuration Item Identifier)
	Generate and Display Probability of Failure
	Validate Entered and Modified Fault Tolerance Data
IT&V Test Utilities Tool	
	Create Test Utilities that may be Invoked by More than One Test, Test Script, or Test Procedure
	Specify Test Environment Including all Test Resources For Each Test
	Test Resources Include:
	Test Plan(s)
	Test Specification(s)
	Test Procedure(s)

SSE Provided Tools	Tool Information/Capability
	Test Scripts
	Test Data
	User Access Profiles
	Test Utilities
	Test Drivers
	Test Stubs
	Test status
	Software Integration Hierarchies
	Testbed Software Configuration
	Test Hardware Configuration Specifications
	Test Results
	Define Test by Test Resource Class
	Define attributes for eachTest Resource Class
	Define Relationships for each Test Resource Class
	Assign Unique ID to a Test Resource
	Provide Capability to Specify Configuration Items to be Tested By a Specific Test in the Test Specification
	Define a Configuration Item Integration Structure : Integration Steps Configuration Items to be Integrated At Each Step, Test Procedure to be Executed at each Step
IT&V Logging Tools (Items that are Logged)	
	Configuration Identification Information of the Test whose Results are being Posted

SSE Provided Tools	Tool Information/Capability
	Status of all Affected Configuration Items Before Testing
	Status of all Affected Configuration Items After Testing
	USERID of the person Posting Results
	Date and Time the Results are Posted
	Optional Remarks at Posting
	USERID of Person Executing the Test Procedure
	USERID of the Person who Authorized The Bypass of the Previous Test in A Test Sequence
	Date and Time of Test Execution
	Configuration Item Identifiers Of the Products being Tested
	Test Results
	Configuration Identification Information of the Test Invoked
	Configuration Identification Information Of the configuration Items Being Tested
	Status of all Affected Configuration Items Before Testbed Build
	Status of all Affected Configuration Items After Testbed Build
	USERID of the Person Building the Testbed
	Date and Time of Testbed Build

SSE Provided Tools	Tool Information/Capability
	Access Control Privileges
	Log Test Resource Status
	Type of Transaction Performed
	User Privileges Before Transaction
	User Privileges After Transaction
	Configuration Identification Information Of the Test Resource Affected
	Status of the Test Resource Before the Transaction
	Status of the Test Resource After the Transaction
	USERID of the Person Performing the Transaction
	Date and Time of the Transaction
	Status of the Configuration Item Before Modification
	Status of the Configuration Item After Modification
	USERID of Person Performing the Modification
	Date and Ti me of the Modification
	Reason for the Modification

SSE Provided Tools	Tool Information/Capability
	The Unique Identifiers for Each Test Resource Specified to be Utilized in each Test Configuration
	Test Resources
	Test Information
IT&V Test Specification Tool	
	Specify Test H/W Configuration
	Specify Test S/W Configuration
	Specify the Test Sequence
	Bypass a Test in Sequence
	Specify Test Utilities for a Test
IT&V User Access Tool (IT&V Users Access may be limited to one or more of the Following)	
	Generate Reports
	View Test Resources and Configuration Items
	View Tranaction Records
	Build Testbeds
	Post Test Results
	Automatically Receive Test Status Messages

SSE Provided Tools	Tool Information/Capability
	Create, Modify, Purge, Delete Test Resources
	Ass, Modify, and Delete IT&V Element Privileges for Specific USERIDs
	Test Initiation
	Enforce User Privileges by Test
	Enforce User Privileges Configuration Identification
	Enforce User Privileges by Test Utility
	Enforce User Privileges by Test Report
IT&V Test bed Tool (Create, Manage, Log Data Associated with the Testbed)	
	Configuration Items to be Tested
	Test Scripts
	Test Procedures
	Test Data
	Test Stubs
	Test Drivers