A PROTOTYPE TO AUTOMATE THE VIDEO SUBSYSTEM ROUTING FOR THE VIDEO DISTRIBUTION SUBSYSTEM OF SPACE STATION FREEDOM

Final Report
NASA/ASEE Summer Faculty Fellowship Program--1993
Johnson Space Center

Prepared By:	Jessie M. Bethly Betz, M.S.
Academic Rank:	Assistant Professor
University & Department:	Southern University & A&M College Department of Computer Science Baton Rouge, Louisiana 70813
NASA/JSC	
Directorate:	Engineering
Division:	Communications and Tracking
Branch:	Systems Engineering
JSC Colleague:	Sally D. Stokes, E.E.
Date Submitted:	August 6, 1993
Contract Number:	NGT-44-001-800

ABSTRACT

The Video Distribution Subsystem (VDS) for Space Station Freedom provides onboard video communications. The VDS includes three major functions: external video switching; internal video switching; and sync and control generation.

The Video Subsystem Routing (VSR) is a part of the VDS Manager Computer Software Configuration Item (VSM/CSCI). The VSM/CSCI is the software which controls and monitors the VDS equipment. VSR activates, terminates, and modifies video services in response to Tier-1 commands to connect video sources to video destinations. VSR selects connection paths based on availability of resources and updates the video routing lookup tables. This project involves investigating the current methodology to automate the Video Subsystem Routing, and developing and testing a prototype as "proof of concept" for designers.

INTRODUCTION

The Johnson Space Center (JSC) is responsible for providing, maintaining, and operating a safe, reliable, and effective National Space Transportation System (NSTS). Further, JSC has significant responsibility for achieving a permanent manned presence in space. This involves creating and operating the Space Station Freedom(SSF) and satisfying a myriad of complex requirements involved in its concept, design and development.[1] The conceptual phase of SSF is complete along with the design and redesign phase, but the total requirements have not been produced for the Video Distribution Subsystem VSR software . VDS provides onboard video communications to SSF. The VDS includes three major functions:

1. External video switching;
2. internal video switching; and
3. sync and control generation.

The external video switching will locate four external video cameras to fourteen locations, controlled remotely through sync and control interface output. The internal video switching will provide viewing for one fixed camera that can be attached to seven internal fixed assemblies. The external camera status is read by status reader in the external switches, and internal camera status by internal switch readers. The internal sync and control generation will allow for a video camera command and status based on the NSTS camera concept.

'The VDS Subsystem Manager Computer Software Configuration Item (VSM/CSCI) is the software which controls and monitors the VDS equipment. It executes in the Standard Data Processor (SDP) and utilizes standard Data Management System (DMS) networking hardware and software. VSM/CSCI operates in conjunction with other customer furnished CSCIs (See Figure 1).

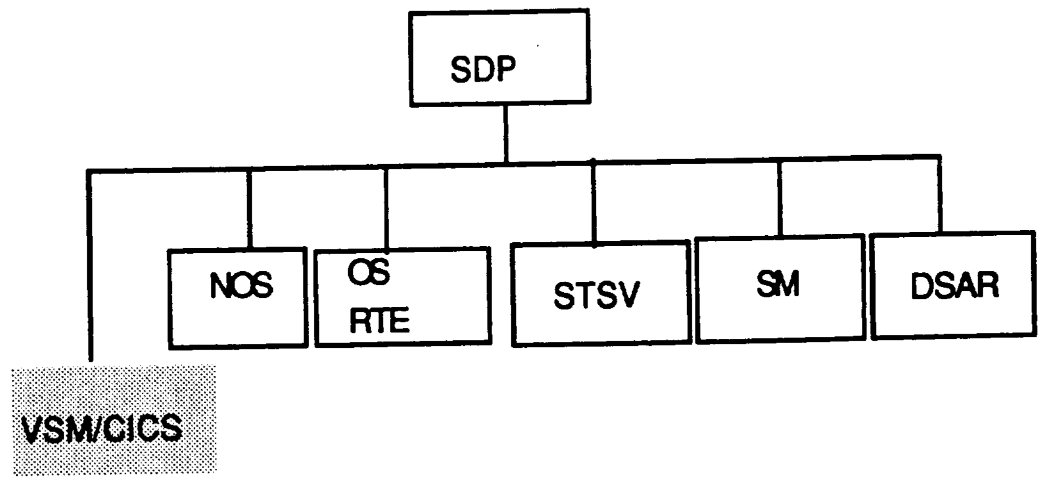

Figure 1.- SDP Overview

NOS:	Network Operating System
OS/RTE:	Operating System/Ada Runtime Environment
STSV:	Standard Service
SM:	System Management
DSAR:	Data Storage and Retrieval

The VSM/CSCI has three major subcapabilities operating under it:
VDS Subsystem Routing (VSR),
External Video ORU Management (EVM), and
Camera Control Input Management (CCIM). (See Figure 2)

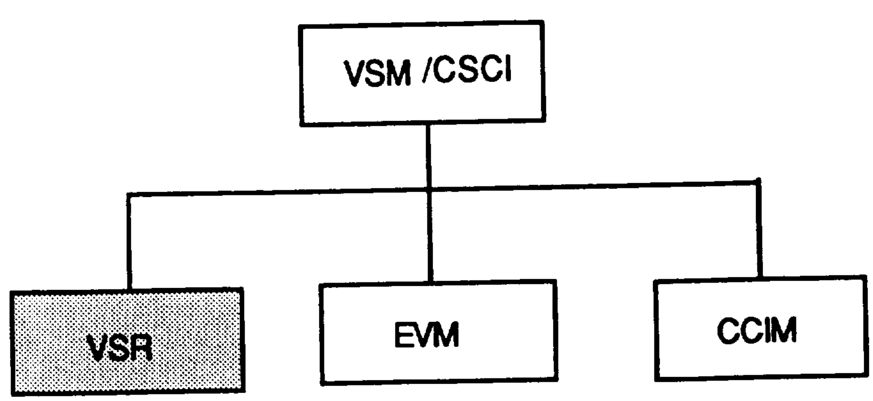

Figure 2.- VSM/CSCI Overview

The Video Subsystem Routing is a part of the VDS Manager Computer Software Configuration Item (VSM/CSCI). It is the responsibility of VSR to activate, terminate, and modify video services in response to Tier-1 commands to connect video sources to video destinations. Teir-1 is the comprehensive term for the coordinated operational command, control, and management of the Space Station. Tier-1 is composed of three major portions:

1. the on-board Integration Station Executive (ISE),
2. the Crew Interface, and
3. the Space Station Control Center. [2].

VSR has the responsibilities of calculating the switch involvement from selected end points; choosing the "best path" from available paths; updating bit strings to reflect the latest selection; and issuing the commands to connect the video source to video destinations. The VSR capability utilizes the following subcapabilities: Command Validation (CV), Video Service Activation (VSA), Video Line Addition (VLA), Video Line Deletion (VLD), Video Service Termination (VST), and Video Table Update (VTU). (See Figure 3).

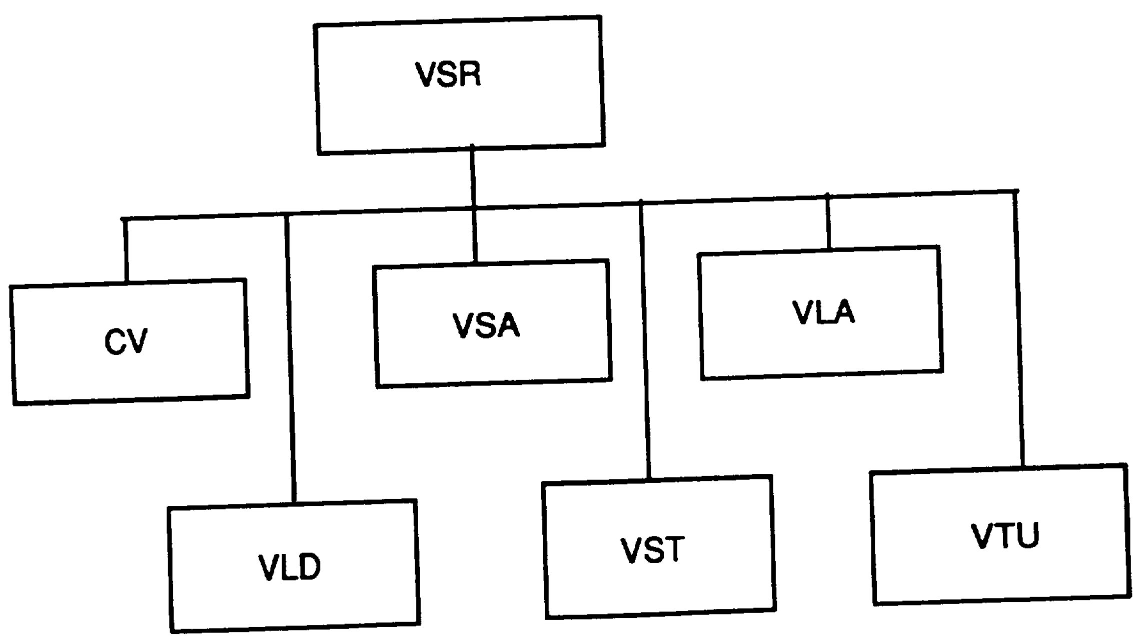

Figure 3.- Video Subsystem Routing (VSR) Capabilities Overview

Command Validation (CV) is the initial receiver for all video commands issued by Tier-1. Each command is checked for syntactic and semantic correctness, then forwarded to the appropriate subcapability to ensure that requested video service is provided. Manual External Video Switch commands are handled exclusively by CV.

Video Service Activation (VSA) accepts Tier-1 initiated Activate Video Service commands from the CV and determines the availability of the requested resources to avoid any conflict. VSA notifies such conflict to Tier-1 by generating a caution and warning advisory message. Once the VSA verifies the resource availability, it creates new video routing paths. If a camera is used as a video signal source, the VSA will initiate a sync connection between the camera and a VSW equipment that hosts the camera. Upon receipt of trunkline status change of the RODB-based VSW Trunkline tables, the VSA updates RODB-based Video Routing Lookup tables.

Video Line Addition(VLA) supports Tier-1 initiated and CV validated Modify Video Service command (Modification Type: Add) which adds another video destination device to an on-going video service without terminating. The Modify Video Service command is passed by CV after validating its semantics and syntactic correctness.

Video Line Deletion(VLD) supports Tier-1 initiated and CV validated Modify Video Service command (Modification Type: Delete) which deletes a video destination device (e.g., video monitor) from an on-going video service without terminating.

Video Service Termination (VST) accepts Tier-1 initiated and CV validated Terminate Video Service command to terminate a video service.

Video Table Update (VTU) updates the RODB-based Video Routing Lookup table, which maintains the source-to-destination video routing path status, by examining the Internal, External or Inter VSW Trunkline tables. VTU also updates the Inter VSW Trunkline table to maintain the VSW trunkline availability status between any Internal and External VSW Orbital Replaceable Unit (ORU) pair.

A prototype to automate the VSR was investigated with emphasis on the Ada programming language and the top-down structured approach to verify the requirements set forth by the Communications and Tracking Systems Division (TCD) of JSC as found in the Flight System Software Requirements Document (FSSR.) "The FSSR is used by the contractor as the basis for the VSM/CSCI design, implementation, and requirements testing, and by the Government and Prime Contractor to assess whether or not the completed VSM/CSCI complies with its requirements."[3].

The automated prototype was not implemented, but an analysis of the requirements were done with recommended revisions, and justification for the use of Ada versus C++ was also investigated. The advantage the author considered was, Ada is a language that embodies and enforces modern software engineering principles. It contains numerous features to support software engineering principles, such as, data abstraction, information hiding, and strong typing, which are presented as secure, reusable software components for large real-time systems as with VSR.

Ada satisfies the following conditions for prototyping. Prototyping is useful in addition to the design phase because it allows the designers to find out whether the system will solve the problem. [4] "The benefits of using a prototype system during the requirements analysis and definition phase of the software life cycle may be summarized as follows:

1. Misunderstandings between software developers and users may be identified as the system functions are demonstrated.
2. Missing user services may be detected.
3. Difficult-to-use or confusing user services may be identified and refined.
4. Software development staff may find incomplete and/or inconsistent requirements as the prototype is developed.
5. A working, albeit limited, system is available very quickly to demonstrate the feasibility and usefulness of the application to management.

6. The prototype may serve as a specification for the development of a production quality system. " [5]

DISCUSSION:

The system development life cycle methodology applied to the investigation of the prototype was divided into two phases:

- traditional: designing hierarchy charts and pseudocode; and
- non-traditional: utilizing the state-of-the-art method of software engineering (computer aided software engineering tool).

In both methods, the top-down structured methodology was applied and hierarchy charts were developed from the requirements document (FSSR). From these design techniques the traditional pseudocode was written to be translated, after NASA redesign, into Ada code for a Unix Sun Workstation. In the non-traditional method, a software engineering package [6] was reviewed to develop models of Ada application systems using graphic icons that may be used to convert the model to the semantics of the Ada programming language utilizing the Ada Structure Graph Editor. The teamwork/Ada notation is derived from the graphic notation designed for Ada by Dr. R. J. A. Buhr. Buhr supports the structured design feature where the notation provides a one-to-one mapping between a set of graphic elements and the corresponding features of the Ada language. Buhr uses Ada structure graphs (ASGs) instead of conventional program structure charts to model relationships that are specific to Ada systems.

The analysis of the requirements design for both methods of the algorithm, help to modify the subcapabilities. The subcapabilities included such things as, repetition of functions in more than one subcapability, as well as some modules with only one or two lines of instructions.

RECOMMENDATIONS:

The redesign of SSF to Space Station Alpha (SSA) has begun, and the DMS (a VDS software interface) has been eliminated, as well as other software interfaces, thereby causing necessary modification or redesign of VSR. Once SSA requirements have been released, it is suggested another requirements analysis and prototype be considered. It is imperative that there exist cameras on the Space Station, but how many, how and where they will be located still needs to be addressed. It is suggested that a software engineering package be considered for prototyping versus using the traditional method of coding. This will reduce the amount of time spent on tedious work versus productivity. It is also suggested that Cadre Technologies, Inc. be considered because of the versatility in software engineering. Cadre Technologies, Inc. also supports the creation of real-time system matrices in the form of two packages: State/Event Matrix(SEM) and Process Activation Table (PAT). The SEM maps states to the events which cause a transition from that state. [6]. The PAT is a matrix that presents the rules, or combinatorial logic, by which events enable and /or disable data flow diagrams(DFD). [6] These two processes can also be utilized in the prototyping of the video routing mappings, video routing status table, and various other tasks to activate the video services.

SUMMARY:

Although a working prototype was not completed, much analysis and redesign has been done to enhance the redesign efforts of the SSA team. A draft of the redesign was provide with modifications to the necessary capabilities. The VSR is one of the most important piece of software for SSF and careful consideration should be given to its design and functions.

REFERENCES:

[1] "Johnson Space Center and the University Community: Teamwork for Space Exploration", 1993, page 2,

[2] Flight System Software Requirements (Communications & Tracking System Software Part 1 - Level B) McDonnell Douglas Space Systems Company Space Station Division DR SY-34-1. WP-2. Contract NAS 9-18200. SSP 30606, Vol II Revision A. January 8, 1993. Page 1.4.

[3] Flight System Software Requirements (FSSR). page 3.2.5-5 SSP 300606, Volume II Revision B.

[4] Pfleeger, Shari and Lawrence. Software Engineering, The Production of Quality Software. Macmillan Publishing Company. 1991. 2nd edition. pages 191-192.

[5] Sommerville, I., Software Engineering, 2nd edition, p.36 Addison-Wesley Publishing Company.

[6] teamwork/Ada: User's Guide. Release 4.0 , 1990. Cadre Technologies, Inc. Providence, Rhode Island. pages 1-2.