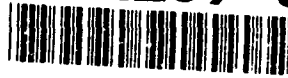AD-A257 655

||||||||||||||||||||||||||||||||

RL-TR-92-101
In-House Report
August 1992

INTRODUCING AUTOMATED SUPPORT FOR MIL-STD-1521B USING AN ADVANCED PROJECT MANAGEMENT SYSTEM

DTIC
S ELECTE D
NOV 1 8 1992
C

James R. Milligan

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

92-29704
||||||||||||||||||||||||||||||||

Rome Laboratory
Air Force Systems Command
Griffiss Air Force Base, New York

7

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-92-101 has been reviewed and is approved for publication.

APPROVED:

SAMUEL A. DINITTO, JR., Chief
Software Technology Division

FOR THE COMMANDER:

JOHN A. GRANIERO
Chief Scientist
Command, Control & Communications Directorate

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE August 1992	3. REPORT TYPE AND DATES COVERED In-House Mar 91 - Aug 91

4. TITLE AND SUBTITLE INTRODUCING AUTOMATED SUPPORT FOR MIL-STD-1521B USING AN ADVANCED PROJECT MANAGEMENT SYSTEM	5. FUNDING NUMBERS PE - 62702F PR - 5581 TA - 18 WU - 47
6. AUTHOR(S) James R. Milligan, Deborah A. Cerino, Arturo B. Ferrer	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Rome Laboratory (C3CB) Griffiss AFB NY 13441-5700	8. PERFORMING ORGANIZATION REPORT NUMBER RL-TR-92-101

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Rome Laboratory (C3CB) Griffiss AFB NY 13441-5700	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Rome Laboratory Project Engineer: James R. Milligan/C3CB (315) 330-2054

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This technical report describes the modeling of MIL-STD-1521B (the Military Standard for Technical Reviews and Audits for Systems, Equipments, and Computer Programs) using a Macintosh-based, Commercial Off-The-Shelf (COTS) Project Management software application. The application, MacProject II (a product of Claris Corporation), is an integrated tool of Rome Laboratory's Software Life Cycle Support Environment (SLCSE) Project Management System (SPMS).

14. SUBJECT TERMS Project Management, SLCSE, Productivity, MIL-STD-1521B, SPMS, Automation, Software Life Cycle, Modeling, CASE	15. NUMBER OF PAGES 28
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT SAR

TABLE OF CONTENTS

LIST OF APPENDICES

DTIC QUALITY INSPECTED 4

i

1. INTRODUCTION

This technical report describes how an advanced computer-based project management system has been augmented to provide automated support for MIL-STD-1521B (the Military Standard for Technical Reviews and Audits for Systems, Equipments, and Computer Programs). The project management system, called the Software Life Cycle Support Environment (SLCSE) Project Management System (SPMS), was developed under a research and development contract sponsored by Rome Laboratory, Griffiss Air Force Base, New York. The system was augmented to provide automated support for MIL-STD-1521B under Rome Laboratory's Advanced System Engineering Concepts in-house program.

This report will primarily be of interest to readers who fall into one of the two following categories:

- **Project Management** personnel (e.g., contractor program managers) who are responsible for planning the project, and then monitoring progress to determine what adjustments, if any, are required to the initial plan. Project managers, in general, view the project in terms of the work tasks that must be completed to achieve project goals, which involves key technical decisions, the assignment of resources (e.g., people, equipment, facilities, etc.) to tasks, cost analyses, risk assessment, etc.

- **Acquisition Management** personnel (e.g., Government program monitors) who are responsible for evaluating the progress of the project with respect to the sponsor's goals. While they are interested in the plan developed by contractor program managers, they tend to view the project in terms of progress toward deliverables, schedules, and overall costs.

2. MIL-STD-1521B AND THE NEED FOR AUTOMATED SUPPORT

DOD-STD-2167A (the Military Standard for Defense System Software Development) defines the various life cycle phases of a software development project as:

- System Requirements Analysis
- System Design
- Software Requirements Analysis
- Preliminary Design
- Detailed Design

- Coding and CSU Testing
- CSC Integration and Testing
- CSCI Testing
- System Integration and Testing

Each phase consists of various tasks, and results in products (e.g., documentation, source code, test cases, etc.). A number of technical reviews and audits are interspersed throughout the various phases, and are special tasks intended to provide a disciplined means of assessing the progress of a project with respect to its objectives and requirements. Each review/audit is designed to address those issues most pertinent to the phase(s) that it follows, and most critical to the successful transition to the next phase.

MIL-STD-1521B defines, in detail, the tasks required for each technical review and audit. The following list identifies each review/audit defined and the life cycle phase that it follows, as defined in DOD-STD-2167A.

Review/Audit	Life Cycle Phase
System Requirements Review	System Requirements Analysis
System Design Review	System Design
Software Specification Review	Software Requirements Analysis
Preliminary Design Review	Preliminary Design
Critical Design Review	Detailed Design
Test Readiness Review	CSC Integration and Testing
Functional Configuration Audit	CSCI Testing
Physical Configuration Audit	CSCI Testing
Formal Qualification Review	CSCI Testing

A large number of tasks are defined by MIL-STD-1521B for each review/audit, as shown below.

Review/Audit	Approximate Number of Tasks
System Requirements Review	26
System Design Review	98
Software Specification Review	14
Preliminary Design Review	265
Critical Design Review	214
Test Readiness Review	13
Functional Configuration Audit	41
Physical Configuration Audit	95
Formal Qualification Review	10

2

To ensure that MIL-STD-1521B was flexible enough to accommodate the needs of a variety of projects, the dependencies between the numerous tasks defined for each review/audit were weakly defined. While the standard's flexibility is advantageous and necessary, it also introduces complexity in planning for and verifying the accomplishment of all the mandated tasks. The number of dependencies for any given project will undoubtedly be very large, and the nature of the dependencies will differ from one project to another, particularly when tailoring is applied to the standard.

The need arises to have some way to specify, capture, and track the dependencies between the tasks required by MIL-STD-1521B for a particular project, such that: (1) a complete plan for each review/audit is provided, (2) the analysis and assessment of the completeness and reasonableness of each plan is possible, (3) the adjustment (e.g., tailoring) of tasks and their dependencies is possible without causing inadvertant negative impacts on other tasks and requirements, and (4) each plan can be tracked so that the accomplishment of its tasks can be recorded and verified.

This need is one ideally suited for an automated solution. Such a solution has been implemented through the adaptation of capabilities provided by SPMS.

3. THE BENEFITS OF AUTOMATED SUPPORT FOR MIL-STD-1521B USING SPMS

As a result of Rome Laboratory's in-house program, SPMS capabilities now include automated support for MIL-STD-1521B. Modeled after the requirements of the standard is a template that contains all of the information for tasks to be performed during the technical reviews and audits required over the course of a DoD-STD-2167A-compliant software development project. This template is designed to allow a project manager/monitor to easily tailor MIL-STD-1521B (without having to refer to a paper-bound copy of the standard) according to the particular needs of any given project, and to establish the dependencies between tasks within a technical review or audit in the most appropriate manner. In addition, the template contains a complete task-to-paragraph cross-reference between the template and MIL-STD-1521B.

Using the template, an acquisition manager can define the specific contract requirements with respect to MIL-STD-1521B and deliver those requirements to the contractor in digital format. The project manager of the

3

contracted organization can then plan the effort precisely according to the specified requirements, and adjust the plan during the performance of the effort within the bounds of those requirements. Digital data interchange allows project assessment by both the acquisition and project managers. Having the various facilities of SPMS at the disposal of the contractor enables the automatic production of the Work Breakdown Structure (WBS) and Organizational Breakdown Structure (OBS) for the effort, as well as a variety of other useful reports and graphics (e.g., activity network diagrams, Gannt charts, and reports on cost, schedule, resources, etc.) that could even serve as deliverables to the acquisition manager.

The benefits of automated support for MIL-STD-1521B using SPMS are summarized as follows:

• A complete, project-specific plan for each MIL-STD-1521B review/audit can be established.

• The completeness and reasonableness of review/audit plans can be more easily assessed.

• Negative impacts resulting from plan adjustments are reduced.

• Review/audit plans can be more easily tracked, recorded, and verified for complete task accomplishment.

• Complete task descriptions for each MIL-STD-1521B review/audit are provided by the template.

• Task-to-paragraph cross-reference between MIL-STD-1521B template and MIL-STD-1521B standard is provided.

• MIL-STD-1521B is easier to understand using the template.

• MIL-STD-1521B is easier to tailor using the template.

• Digital data interchange versus paper interchange of data (e.g., contract requirements, management plans, etc.) for MIL-STD-1521B-compliant projects between the contracting agency and the contractor is supported.

• Automatic production of various project management reports, charts, and diagrams showing multiple views of MIL-STD-1521B-compliant project data is provided.

• Merger of MIL-STD-1521B task information with a complete life cycle database (i.e., the SLCSE DOD-STD-2167A database).

4. THE MODELING OF MIL-STD-1521B

In modeling MIL-STD-1521B, a high-level activity network was devised for DOD-STD-2167A. This activity network includes the tasks (i.e., activities, such as Software Requirements Analysis), milestones (i.e., products, such as the Software Requirements Specification), and subprojects (i.e., composite activities, such as the Software Requirements Review, that consist of finer-grained activities at a more detailed, lower-level of abstraction).

Thus, each technical review and audit defined by MIL-STD-1521B was modeled as a subproject that contains all of the various tasks mandated by the standard. Each subproject, in turn, is capable of containing its own subprojects for tasks comprised of even finer-grained activities, and so forth. This nesting of subprojects permits the logical grouping of tasks in much the same way that MIL-STD-1521B does so using nested paragraphs.

MacProject II, a Commercial-Off-The-Shelf (COTS) software product by Claris Corporation that is an integral tool of SPMS, was used to implement the activity networks. These networks form a template which can be duplicated and then tailored to the specific requirements of a particular project. In digital form, MIL-STD-1521B can be tailored by the deletion of non-applicable tasks and the definition of task dependencies. For convenience, each task description was supplied with a subtitle that references the corresponding paragraph of MIL-STD-1521B. From the activity networks template, a variety of other reports, charts, and graphs can also be generated by SPMS for project planning, assessment, and adjustment.

5. CONCLUSION

Implementation of the MIL-STD-1521B template provides a powerful tool to both acquisition and project managers. Combined with SPMS capabilities, the potential benefit of applying the template to software development projects is magnified by several orders.

6. REFERENCES

[1] DOD-STD-2167A, Military Standard, Defense System Software Development, February 1988

[2] MacProject II User's Guide, 1991, Claris Corporation, Santa Clara, CA

[3] MIL-STD-1521B, Military Standard, Technical Reviews and Audits for Systems, Equipments, and Computer Programs, June 1985

[4] SPMS, Software Design Document, August 1991, (CR-4-1606), General Research Corporation, P.O. Box 60, Santa Barbara, CA

[5] SPMS, Software Development Plan, June 1990, (CR-6-1606), General Research Corporation, P.O. Box 60, Santa Barbara, CA

[6] SPMS, Software Requirements Specification, December 1990, (CR-5-1606), General Research Corporation, P.O. Box 60, Santa Barbara, CA

[7] SPMS, System/Segment Design Document, October 1990, (CR-7-1606), General Research Corporation, P.O. Box 60, Santa Barbara, CA

[8] J. Milligan, Software Life Cycle Support Environment (SLCSE) Project Management System (SPMS): Not Just Another Project Management Tool, July 1989, Griffiss AFB, NY

[9] F. LaMonica, Software Life Cycle Support Environment (SLCSE), Product Overview, SLCSE Enhancements & Demonstration Program, July 1991, Griffiss AFB, NY

[10] Software Life Cycle Support Environment, Software Detailed Design Document, July 1989, (CR-7-1537), General Research Corporation, P.O. Box 60, Santa Barbara, CA

[11] Software Life Cycle Support Environment, Software Requirements Specification, November 1988, (CR-3-1537), General Research Corporation, P.O. Box 60, Santa Barbara, CA

[12] Software Life Cycle Support Environment, Software Top Level Design Document, November 1988, (CR-4-1537), General Research Corporation, P.O. Box 60, Santa Barbara, CA

[13] Software Life Cycle Support Environment, Software User's Manual, August 1989, (CR-6-1537), General Research Corporation, P.O. Box 60, Santa Barbara, CA

[14] T. Strelich, Software Life Cycle Support Environment, Final Technical Report, February 1990, RADC-TR-89-385, Rome Air Development Center, Griffiss Air Force Base, NY

APPENDIX A - Software Life Cycle Support Environment (SLCSE)

SLCSE (pronounced "slice") is a computer-based framework for the instantiation of Software Engineering Environments (SEEs) that are tailored to accommodate the specific needs of software development projects. A SEE instantiated from the SLCSE framework consists of a set of integrated tools that support the Full-Scale Development (FSD) and Post-Deployment Software Support (PDSS) phases of the Mission Critical Computer System (MCCS) software life cycle. Specifically, SLCSE provides support for the various phases and inter-phase activities of the software life cycle, including requirements specification and analysis, design, coding, unit/integration testing, Quality Assurance (QA), Verification and Validation (V&V), project management, and Configuration Management (CM).

As illustrated in Figure B-1, the SLCSE top-level architecture consists of four major subsystems: the User Interface, Command Executive, Database, and Toolset.

The User Interface is window-oriented and menu-driven, providing a common and consistent style of operation to all its users. While the User Interface is consistent in style, the tools and database views a user has access to are governed by the various "role(s)" the user has been assigned by the framework administrator.

The Command Executive controls all SLCSE functions (apart from those which are specific to a tool). It interfaces with the user through the User Interface, and invokes tools in the Toolset.

The Database provides SLCSE applications with an Entity-Relationship (ER) InterFace (ERIF) to an underlying relational database engine. At its highest level of abstraction, the SLCSE Database appears as a single expansive ER network of information. The current ER database schema models the data requirements of DoD-STD-2167A (the Military Standard for Defense System Software Development), and is so comprehensive and complete that each of the seventeen (17) DoD-STD-2167A data items (i.e., documents and specifications) can be automatically generated.

The Database is the most critical and important part of SLCSE. It serves not only as a repository for formal life cycle information required by DOD-STD-2167A (or potentially by any other life cycle model), but also as an integrating mechanism for tools by allowing them to share information. As

SLCSE TOP-LEVEL ARCHITECTURE

**Multiple
User Roles**

Acquisition Management
Project Administration
Project Management
System Analysis
SW Analysis
Programming
SW Testing
SW Integration
System Integration
Verification & Validation
Quality Assurance
Configuration Management
Post Deployment SW Support
Training
MCCS Operation
SLCSE Installation
Framework Administration
Secretarial
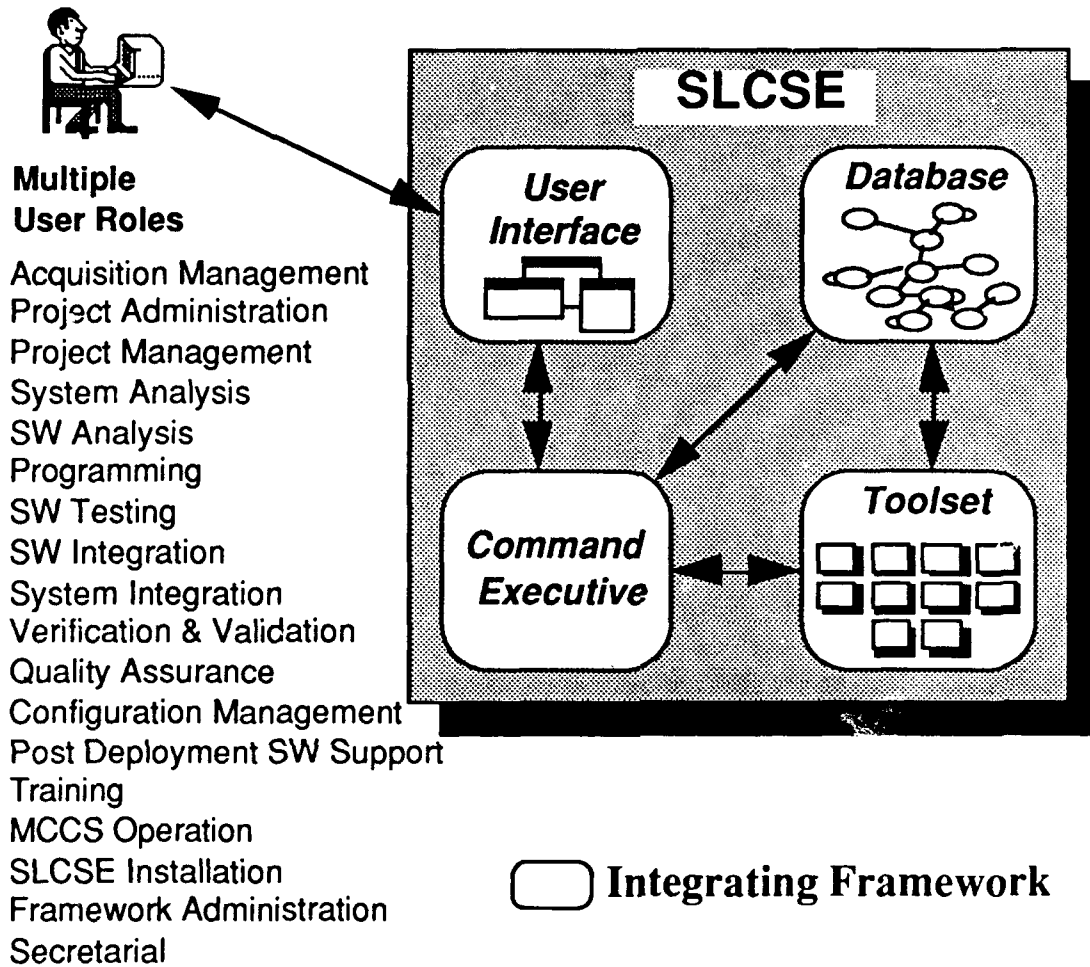
⬜ **Integrating Framework**

Figure B-1

data is generated and stored by users and tools, it becomes available through
the Database services for use by other users and tools in subsequent activities
and life cycle phases. The SLCSE integrating framework, and in particular,
the interaction between tools and the Database, is what makes possible many
life-cycle oriented technological opportunities and potential productivity
gains.

The User Interface, Command Executive, and Database are considered
to be the "integrating framework" of SLCSE, and are the three subsystems
with which tools of the Toolset are integrated. A tool is invoked by the
Command Executive from a tool user interface that may be "conformant" to
the SLCSE User Interface, and may interact with other tools through the

Database, which stores and manages all of the project data created throughout the life cycle.

SLCSE has been designed to allow virtually any number of tools to be integrated into the Toolset to support a variety of software engineering methods and overall development methodologies. This "methodology independent" tool integration concept provides for the use of Commercial Off-The-Shelf (COTS) tools, as well as tools developed specifically for use within SLCSE.

APPENDIX B - SLCSE Project Management System (SPMS)

The SLCSE Project Management System (SPMS) is a recent extension of SLCSE that provides advanced project management capabilities.

The SPMS operational concept centers around making it possible for high-productivity Macintosh-based project management tools to exchange information with the SLCSE Database in much the same manner that VAX-based SLCSE tools do. As previously mentioned, the component that makes possible many technological opportunities and potential productivity gains is the SLCSE Database. While the SLCSE Database provides a repository for the formal life cycle information required by DoD-STD-2167A, SPMS serves as a system that populates portions of the DoD-STD-2167A database with information generated by high-quality COTS project management tools. In particular, SPMS populates entities (e.g., *Activity* entities) and relationships (e.g., *Activity Depends_On Activity* relationships) of the SLCSE Database with data maintained by such tools as MacProject II (e.g., the tasks and task dependencies of a graphically displayed activity network).

Figure B-1 depicts the architecture of SLCSE as augmented by SPMS. Shaded boxes denote those components of SLCSE which existed before SPMS was developed. SPMS augments SLCSE with a client/server architecture that makes it possible for remote applications (in the case of SPMS, Macintosh-based project management tools) to import over a network a subset of information contained in the SLCSE Database, manipulate that local subset of information (in the case of SPMS, information stored in the Macintosh SPMS Database), and then export the modified information back to the SLCSE Database. Consequently, this information (in the case of SPMS, information that is primarily maintained in the Project Management subschema of the DoD-STD-2167A SLCSE Database) can be made available to authorized users through any tool that is integrated with SLCSE regardless of the tool's host computer.
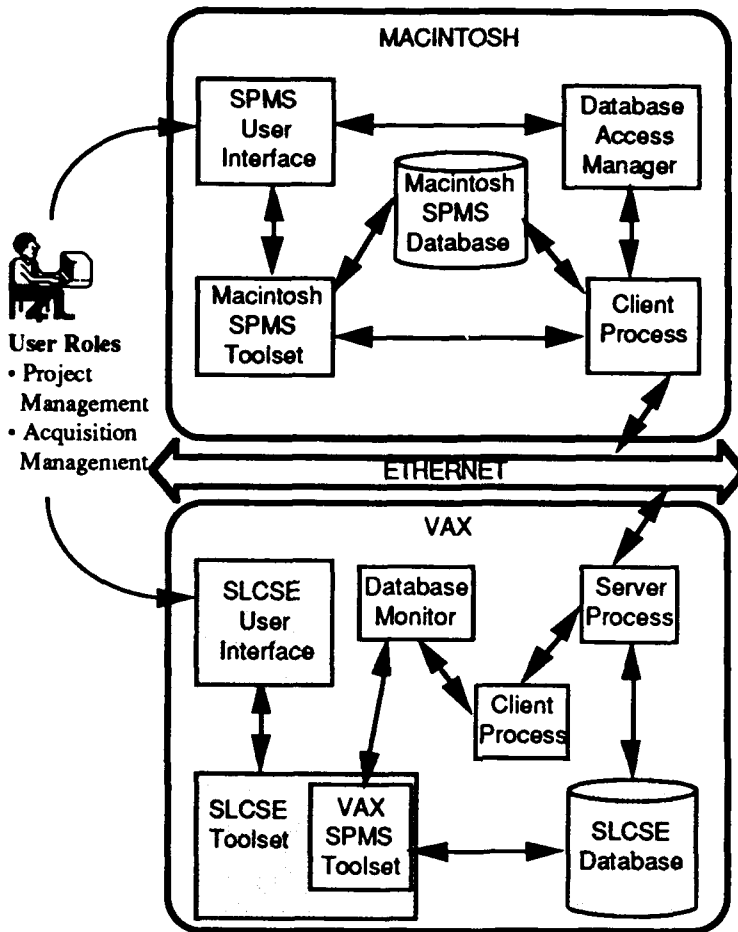
SLCSE/SPMS Architecture

Figure B-1

SPMS provides many of its project management capabilities through the use of COTS tools hosted on the Macintosh, since these tools are easy to use, highly functional, quite affordable, and noted for their high-quality graphic outputs. In addition to Macintosh-based capabilities, a VAX SPMS Toolset, and a VAX-based Database Monitor are included as part of SPMS.

As indicated in Figure B-1, the needs of two types of users are supported by SPMS. In SLCSE terms, the two user roles supported are:

- **Project Management** personnel (e.g., contractor program managers) who are responsible for planning the project, and then monitoring progress to determine what adjustments, if any, are required to the initial plan. Project managers, in general, view the project in terms of the work tasks that must be completed to achieve project goals, which

involves key technical decisions, the assignment of resources (e.g., people, equipment, facilities, etc.) to tasks, cost analyses, risk assessment, etc.

- **Acquisition Management** personnel (e.g., Government program monitors) who are responsible for evaluating the progress of the project with respect to the sponsor's goals. While they are interested in the plan developed by contractor program managers, they tend to view the project in terms of progress toward deliverables, schedules, and overall costs.

SPMS operates in one of two modes, depending on the status of the communication path between SPMS components and the SLCSE Database: (1) the SLCSE-Coupled Mode, and (2) the Macintosh Stand-Alone Mode.

The **SLCSE-Coupled Mode** exists when SPMS tools are able to communicate directly to the SLCSE Database. The VAX SPMS Toolset always operates in this mode, since there is no provision for the maintenance of a local subset of SLCSE Database information on a remote VAX node. The SLCSE-Coupled Mode also exists when both the Client Process (on the Macintosh) and the Server Process (on the VAX that hosts the SLCSE Database) are active and able to transfer data directly over the network.

The **Macintosh Stand-Alone Mode** exists when the Server Process on the VAX is inactive, and there is no communication path between the Macintosh and the SLCSE Database. This mode of operation allows the user to work from the local Macintosh SPMS Database (e.g., using SPMS on a Macintosh Portable at a location where a connection to the SLCSE Database is not possible), and then export the local subset of information to the SLCSE Database at a later time (in the SLCSE-Coupled Mode).

SPMS comprises a set of project management tools that are available both on the Macintosh and on the VAX. The kinds of tools that are available are:

- **Planning and Adjustment Tools**
  Tools that support the establishment of the project Work Breakdown Structure (WBS), the Organizational Breakdown Structure (OBS), the project schedule, and the identification of resources, costs, activities, and milestones. These same tools are applied over the course of the project to adjust the project plan.

- **Tracking, Monitoring, and Assessment Tools**

    Tools that provide reports and displays of project management information, including the reports associated with event monitoring. In addition to a set of standardized reports and SLCSE Database event monitoring query specifications, a user can design additional reports, specify custom event monitoring, and make ad-hoc queries to the SLCSE Database.

- **Database Manipulation Tools**

    Tools that manage the formatting and transfer of data between tools, the Macintosh SPMS Database, and the SLCSE Database. In particular they: (1) transfer project data from one tool to another, (2) format and transfer project data output by tools into the Macintosh SPMS Database or the SLCSE Database, and (3) transfer and format project data from the SLCSE Database or the Macintosh SPMS Database to the tools.

    SPMS includes both COTS tools and tools developed specifically for SPMS (i.e., developmental software), as listed below:

- **SPMS COTS Tools**

    MacProject II - Claris Corporation
    - Project Management (Macintosh)
    Excel - Microsoft
    - Spreadsheet (Macintosh)
    Word - Microsoft
    - Word Processor (Macintosh)
    Supercard - Silicon Beach Software
    - SPMS User Interface (Macintosh)
    Mac241 - White Pine Software
    - Terminal Emulator (Macintosh)
    Lotus 1-2-3 - Lotus Development
    - Spreadsheet (VAX)

- **SPMS Developmental Software Tools**

    MacProject II Companion Database Interface (CDI) Tool
    - Database Manipulation Tool (Macintosh)
    Excel CDI Tool
    - Database Manipulation Tool (Macintosh)
    Word CDI Tool
    - Database Manipulation Tool (Macintosh)
    Database Access Manager

- Database Manipulation Tool (Macintosh)

WBS Editor (Excel-Based)
- Project Planning/Adjustment Tool
  (Macintosh)

OBS Editor (Excel-Based)
- Project Planning/Adjustment Tool
  (Macintosh)

PCR Tool (Excel-Based)
- Project Tracking/Monitoring/Assessment Tool
  (Macintosh)

Client/Server Components
- Database Manipulation Tool
  (Macintosh/VAX)

Database Monitor
- Event Monitoring Tool (VAX)

WBS Editor (Lotus-Based)
- Project Planning/Adjustment Tool (VAX)

OBS Editor (Lotus-Based)
- Project Planning/Adjustment Tool (VAX)

*SLCSE Problem/Change Report Processor
(PCRP) Upgraded Tool*
- Project Assessment Tool (VAX)

*SLCSE DOCGEN Standardized Project
Management Reports*
- Project Tracking/Monitoring/Assessment
  Automatic Document Generation Tool (VAX)

SPMS provides many special features that are not currently found in other project management tools primarily because it leverages off of, and extends, the capabilities already existing in SLCSE and COTS software. The special features of SPMS are, in summary: (1) its evolution of SLCSE, (2) the automated project monitoring capability it provides within SLCSE, (3) its standard and custom report generation capabilities, and (4) the extensibility of its toolset.

An important feature of SPMS is the evolution of SLCSE to provide a framework for the integration of distributed tools. The basic capability for the interchange of data between the SLCSE Database and tools distributed throughout a heterogeneous network of computers is provided by components that resulted from the development of SPMS. SPMS employs a client/server architecture with a remote application interface to the SLCSE Database (called the High Level Entity-Relationship Interface (HLERIF)).

This helps improve user productivity as well as the performance of the environment by off-loading work from the SLCSE host machine to the workstations naturally preferred by users. At the same time, the advantages of tool interoperability and data sharing (as in the original SLCSE operational concept) are preserved.

Figure B-2 shows the client/server architecture as applied to MacProject II, the primary COTS tool of SPMS. MacProject II provides an import/export capability for the purpose of interoperability with other Macintosh tools (e.g., Excel and Word). In particular, MacProject II information can be exported in the form of an ASCII text file, and subsequently imported by other Macintosh tools, and visa versa. For SPMS, a MacProject II Companion Database Interface (CDI) tool was developed to reformat a MacProject II import/export ASCII text file into a "collection" file. The collection file contains information about a set (or collection) of entities and relationships, and is in a format that may be exported to the SLCSE Database via the client/server architecture. Specifically, a Database Access Manager (DAM) that is coupled to the HLERIF allows the SPMS user to manage the transfer of information between collection files on the Macintosh and the SLCSE Database. CDI tools for the COTS tools Excel and Word were also developed, as both of these tools also possess import and export capabilities.

Another special feature provided by SPMS is the capability to automatically monitor a SLCSE project for the occurrence of specific events. The component of SPMS that helps to automate the project monitoring activity of the project management process is the SLCSE Database Monitor on the VAX.

The SLCSE Database Monitor allows a project manager to establish an "events-of-note" list that is specified in the Document Generation Language (DGL) of SLCSE. The DGL describes the queries to be made to the SLCSE Database for monitoring purposes and the text processing commands required to format the monitoring information gathered. DGL for a monitoring action is contained in an attribute of a *Report* entity in the SLCSE Database. The *Report* entity is linked by a relationship to a *Monitor* entity. The *Monitor* entity contains scheduling and personnel notification specifications that would typically be established by a project manager. The SLCSE Database Monitor then periodically reads *Monitor* entities to determine whether or not it is time to automatically generate reports based on the DGL contained in related *Report* entities. If a report is to be generated, the SLCSE Database Monitor invokes the SLCSE tool

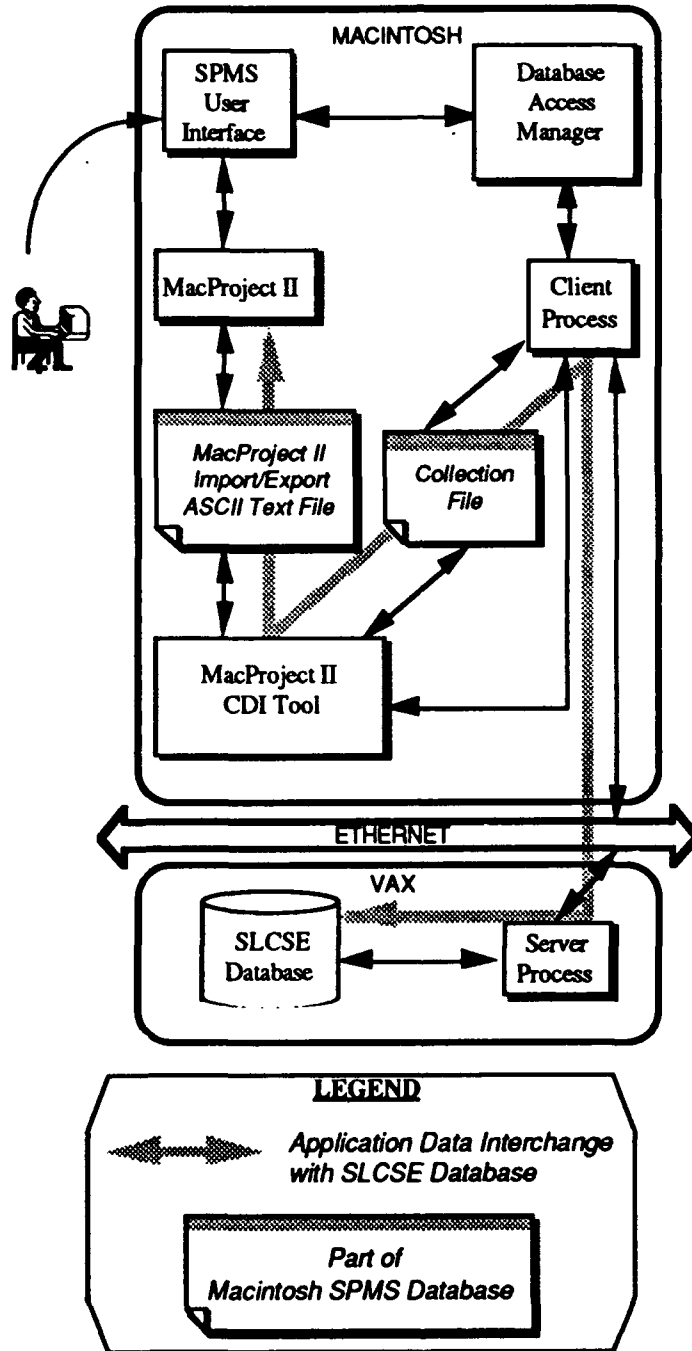## SPMS Revolutionizes SLCSE
With A Client/Server Architecture

Figure B-2

DOCGEN-Report, and then either: (1) electronically mails the resulting report to project personnel specified in the *Monitor* entity, or (2) notifies (via electronic mail) project personnel (specified in the *Monitor* entity) that a

report was generated and where it is located. SPMS includes a number of pre-defined *Report* entity DGL specifications for typically desired monitoring reports. Additional monitoring reports may also be custom designed by the SPMS user.

In addition to the project monitoring reports, SPMS includes pre-defined DGL specifications for the generation of a variety of other project management reports. Like the monitoring reports, these reports are also generated on the VAX from information stored in the SLCSE Database via the DOCGEN-Report tool, and may be customized by modifying the DGL used for report generation. On the Macintosh, Excel macros have been developed to generate (also from SLCSE Database information) a set of standard project management reports that feature sophisticated graphical representations of project information. As with DGL, Excel macros can also be designed by the SPMS user for customized reports.

An essential feature of SPMS is its extensibility in terms of integrating additional project management tools either on the VAX or on the Macintosh. Tools can be integrated within the VAX SPMS Toolset through the SLCSE framework. On the Macintosh, it is possible to add tools to the *Tools* menu of the SPMS User Interface simply by entering the name of the application, and specifying its hard drive location. Once this is done, the new application may be launched directly from the SPMS User Interface. This eliminates the need to halt SPMS operation in order to use an otherwise external application.