NUWC-NPT Technical Report 10,777
3 January 1995

A Framework for Integrating Cost Estimating and Systems Engineering Tools

Thomas C. Choinski
Daniel J. Organ
Submarine Sonar Department

John J. McGahan
Arve Sjovold
Robert W. Thompson
Tecolote Research, Inc.

DTIC
S ELECTE D
FEB 16 1995
B

UNITED STATES NAVY

Naval Undersea Warfare Center Division
Newport, Rhode Island

19950208 010

Approved for public release; distribution is unlimited.

PREFACE

The research described in this report was sponsored by E. Wald of the Office of Naval Research through the *Engineering of Complex Systems Program*, Program Manager Dr. H. Crisp, Naval Surface Warfare Center (NSWC), Division Dahlgren.

The technical reviewer for this report was A. A. Shalhout (Code 33B) of NUWC Detachment New London.

The authors would like to acknowledge the technical direction provided by Steven Howell of NSWC Division Dahlgren and Cuong Nguyen of NSWC Detachment White Oak during the course of this research.

Reviewed and Approved: 3 January 1995

RJ Martin

R. J. Martin
Acting Head, Submarine Sonar Department

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

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
3 January 1995

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
A Framework for Integrating Cost Estimating and Systems Engineering Tools

5. FUNDING NUMBERS
N0001495WXZ0080

6. AUTHOR(S)

T. C. Choinski and D. J. Organ
Submarine Sonar Department

J. J. McGahan, A. Sjovold, and R. W. Thompson
Tecolote Research, Inc., Santa Barbara, CA

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Undersea Warfare Center
Detachment New London
New London, Connecticut 06320

8. PERFORMING ORGANIZATION REPORT NUMBER
TR 10,777

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
800 North Quincy St.
Arlington, VA 22217-5000

Naval Surface Warfare Center
Dahlgren Division
Dahlgren, VA 22448--5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

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

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

As the defense budget continues to decline, the incorporation of cost estimating into the system engineering process has become increasingly important. Although cost estimators and systems engineers are both using automated tools to perform their tasks, the integration of such tools has been limited. This report provides a description of a framework for integrating the cost estimator's tool (called ACEIT) with a set of system engineer's tools (collectively called DESTINATION).

DTIC QUALITY INSPECTED 4

14. SUBJECT TERMS
ACEIT, Cost Engineering Model, Cost Estimating Methods, DESTINATION, SETIS, Systems Engineering Tools

15. NUMBER OF PAGES
30

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

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std Z39-18
298-102

TABLE OF CONTENTS

Accession For
NTIS GRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

ACE	Automated Cost Estimator
ACEIT	Automated Cost Estimating Integrated Tools
ADP	Automated Data Processing
ADPE	Automated Data Processing Equipment
ARC	Avionics Reliability Cost
BBEST	Black Box Estimator
CAIG	Cost Analysis Improvement Group
CASE	Computer-Aided Software Engineering
CER	Cost Estimating Relationship
CES	Cost Element Structure
COEA	Cost and Operational Effectiveness Analysis
CO$TAT	Statistical Analysis for Cost Analysts
DESTINATION	Design Structuring and Allocation Optimization
DoD	Department of Defense
DOS	Disk Operating System
ECS	Engineering Complex Systems
ESC	Electronic Systems Center
IV&V	Independent Verification and Validation
LCC	Life Cycle Cost
MS	Microsoft
NASA	National Aeronautics and Space Administration
NUWCDETNLON	Naval Undersea Warfare Center/Detachment New London
O&S	Operation and Support
PC	Personal Computer
RI$K	System for Cost Estimate Risk Assessment
SDF	System Design Factors
SETIS	Systems Engineering Technology Interface Specification
VHDL	VHSIC Hardware Description Language
WBS	Work Breakdown Structure

A FRAMEWORK FOR INTEGRATING COST ESTIMATING AND SYSTEMS ENGINEERING TOOLS

INTRODUCTION

The need to build complex warfare systems within an industry characterized by a declining defense budget has led to an increased interest in the role of cost modeling during the design stage of system development. Accordingly, to ensure the production of systems that are both cost-efficient and well-designed, the systems engineering process must learn to incorporate cost modeling capabilities. This report describes the framework for integrating cost estimating with systems engineering as conceived by Choinski and Organ.[1] Their approach focuses on two integration tools: DESTINATION and ACEIT.

The first tool, called DESTINATION (design structuring and allocation optimization), was developed within the *Engineering of Complex Systems (ECS) Block Program* managed by the Naval Surface Warfare Center/Division Dahlgren in Dahlgren, Virginia.

The second, referred to as ACEIT (automated cost estimating integrated tools), is an integrated family of products designed to assist in conducting cost analysis activities, such as cost estimating, what-if studies, cost proposal preparation and evaluation, risk and uncertainty analysis, and cost estimating relationship (CER) development. ACEIT was developed under the direction of the Air Force Electronics Systems Center (ESC) in Bedford, Massachusetts.

This report will describe the integration framework by first presenting background material on the systems engineering and cost estimation processes. In the sections that follow, a brief overview of DESTINATION and ACEIT is provided, the preliminary interface specification is outlined, and the integration approach is summarized.

BACKGROUND

To understand the complexities of integrating a general purpose cost estimating capability into a systems engineering methodology, it is necessary to examine the cost estimation and systems engineering processes.

SYSTEMS ENGINEERING OVERVIEW

The systems engineering process consists of the following sequence of steps:

1. Identifying system need,

2 Defining system requirements,

3. Identifying system functions and modes to support requirements,

4. Allocating resources to system functions,

5. Performing tradeoff analysis based on design objectives,

6. Modifying system definition, and

7. Documenting system design.

These steps, which were adapted from MIL-STD-499B and the classic systems architecting waterfall, represent a distinct sequence of events in the evolution of a system from concept to implementation.[2,3] However, in practice, the steps are often eliminated, combined, or reordered to address programmatic, budgetary, or scheduling constraints.

As military systems have become more complex, the responsibilities of the systems engineer have become more difficult. Although improved computer technology has made numerous automated tools available to aid the system engineer during the synthesis of complex systems, each tool focuses only on a particular aspect of systems engineering. For example, computer-aided software engineering (CASE) tools, such as CADRE Teamwork, support the system software design, while tools such as the VHSIC hardware description language (VHDL) primarily address system hardware design. Other tools, such as Network II.5, support efforts only related to the system architecture. Currently, there is significant interest in integrating the capabilities of all these tools so that systems engineers are provided with a "comprehensive environment" for the synthesis of complex systems.

The NSWC/White Oak Detachment scientists involved with the ECS Research Block are currently developing such a comprehensive environment. Collectively called DESTINATION (as described earlier), this environment provides tools, design aids, and methodologies to support design structuring, resource allocation, and optimization capabilities for assisting the systems engineers with their tasks.

In the commercial market, RDD-100, available from Ascent Logic Corporation, is an example of an automated systems engineering tool that addresses all aspects of the system engineering life cycle, including extraction, refinement, and allocation of requirements, simulation and verification of requirements, and documentation of the design and the design process.

COST ESTIMATION OVERVIEW

Similar to the systems engineering process, cost estimation can also be viewed as a sequence of steps:

1. Defining the baseline estimate,
2. Developing the estimate,
3. Conducting sensitivity and "what-if" analyses,
4. Performing cost risk assessment on the estimate, and
5. Documenting the estimate.

Each step is important in the completion of almost any weapons system cost estimate. While the steps presented here are in linear order, the process is very dynamic. Technical, schedule, programmatic, or political pressures can often identify new requirements and dictate changes.

Before cost can be determined, a clear definition of the system to be estimated must be developed. System performance characteristics and programmatic requirements must be understood. A work breakdown structure (WBS) and/or cost element structure (CES) must be developed and each item in the structure clearly explained.

Once the WBS/CES has been determined, the estimate can be developed. This process entails identifying or developing appropriate estimating methodologies for each element. Typical approaches include analogy, expert opinion, bottoms-up engineering, and top-down parametrics. These approaches are further adjusted for burden rates, inflation, and learning curve considerations, as needed. The estimate is phased over time to reflect development, production, and support schedules and budgets.

When the baseline estimate has been established, a "sensitivity" analysis can be performed to identify the primary cost drivers and develop a range of possible outcomes over which the basic estimating methodology is valid. This analysis is critical in helping to identify areas of the estimate that are key to cost risk assessment. What-if analyses can be performed to address alternate baseline scenarios and to estimate possible contingencies.

Realistically, estimating the expense of developing new technologies and their long-term life cycle costs (LCCs) is far from a perfect science. Quantifying the uncertainty in the estimate can be as complex as developing the estimate itself since a cost estimate is usually prepared as a "point" estimate, representing one outcome from a set of possible outcomes.

The process of investigating the uncertainties in a cost estimate is called risk analysis. The results of such investigations are important for project management, budgetary processes, and the evaluation of cost and operational effectiveness analyses (COEAs). It is the purpose of risk analysis to evaluate the estimating methods and the technical and programmatic factors used in developing the estimate and to quantify their effects on costs for presentation to decision makers. The risk assessment casts the point estimate as one possible outcome within a range of possible outcomes and attempts to quantify its likelihood of actually occurring.

COST ENGINEERING MODEL DEVELOPMENT

OVERVIEW

Cost engineering is the common term used to describe the combined efforts of system design engineers and cost estimators in the development of cost estimates for complex military systems. Its effectiveness lies in bringing engineering flexibility and sensitivity to the task of cost estimating as well as cost visibility to the design engineering process. It is especially useful in allowing many tradeoff alternatives to be explored, since it provides for cost to be weighted equally with other system objectives defined by the customer. Too often in the past, the cost analyst was given the task of evaluating the cost of acquiring a system design that had long ago been frozen. The acquired system would be cost effective only to the degree that the designer had been sensitive to the parameters that drive cost. Any insights that the cost analyst might have been able to provide would have been lost due to the unrealistic sequence of the process. For complex systems, it is unfair to ask the designer to take full responsibility for the ultimate costs, especially when full LCCs are included. Clearly, engineers and cost analysts must communicate early in the system design stage of a program if there is to be any hope of developing cost effective systems. The cost engineering approach is intended to provide this capability.

However, identifying what cost engineering is supposed to do does not explain how to create an effective cost engineering capability. Deriving good, competent cost estimates for an alternative design can be done quite well if the system engineer has the time to prepare a detailed technical baseline. Unfortunately, conducting such a labor-intensive effort, in combination with using a wide range of alternatives, would be prohibitively expensive. In addition, the system engineer typically distills the design to such a level of detail that the relationship between cost drivers and performance parameters is often lost. When exploring alternatives, the system engineer must understand which parameters and requirements have the most effect on cost. Similarly, the cost estimating methodologies must provide links that clearly relate system design parameters to their expense.

It should be noted that there are a number of steps to developing cost engineering models that can be easily integrated into the systems engineering environment. However, with the advanced technologies being introduced into military systems so rapidly, many cost databases and methodologies have become outdated. Developing parametric CERs based on past systems may not prove very helpful. Therefore, the engineer must be able to describe to the cost analyst the features of an advanced technology that are likely to drive its cost. If the cost analyst can be made to understand the principles influencing cost for a particular technology, it should be possible to construct useful and broadly applicable cost engineering models. This was demonstrated in the development of the Tecolote Research, Inc., avionics reliability cost (ARC) tradeoff model. ARC was designed to enable a system designer to estimate costs of new configuration processors incorporating advanced technology integrated circuits.

Another stage in the development of cost engineering capabilities requires the minimization of system LCCs. Much has been written concerning the importance of addressing LCC in system design, but often too little attention was paid to the costs of deployment and ownership. A system might be designed to be cost effective based on the tradeoffs between development and production, but it might still be extremely expensive to operate and maintain. It has become quite clear that the parameters that determine systems operation and support (O&S) costs are typically established early in the design process, mostly by default. It takes a concerted effort on the part of system engineers to incorporate O&S parameters in the initial design tradeoffs. A carefully prepared cost engineering model can provide such a capability by addressing those technical features of the design that affect reliability and maintenance action. It is also possible in the early stage of the design process to consider features of the logistic support system as candidates for tradeoffs if they can reduce LCCs. For example, as was shown in the ARC model, the choice of advanced technology digital integrated circuits resulted in lower cost because their high reliability permitted use of a two-level maintenance system over the traditional three-level one.

The advent of advanced memory devices in electronic systems has allowed for tremendous growth in the amount of software that can be embedded within a military system. Quite often, in new military systems, the software embodies a majority of the system functions. Although such software is expensive to develop, it is essentially free to produce. However, it can be very costly to maintain, accounting for the greatest portion of O&S costs in some newer equipment. Therefore, software maintenance costs, as well as development costs, must be factored into the early system tradeoff analysis if LCC is to be minimized. Cost engineering models must be constructed that are sensitive to the amount and type of embedded software, the amount of independent verification and validation (IV&V) applied, the frequency of upgrades, and the requirements for support and maintenance. As hardware and software become more and more integrated, new cost engineering models must be developed for determining the most appropriate tradeoffs.

The above description examines only some of the reasons why it is important to bring engineers and cost analysts together early in the design process. If cost engineering models can be developed with the proper scope, degree of fidelity, and ease of use, the system design team can begin to configure cost-effective military systems. However, the task of developing a cost engineering model is not easy. Such models must be applicable not only very early in the design process, as previously discussed, but also later when the design is further developed. Past experiences with the development of models such as ARC suggest that this is feasible for carefully selected sets of hardware systems. Since cost engineering models must by nature incorporate a great deal of engineering expertise, it is very important to thoroughly define the class of systems that will be addressed. As the scope of the model is narrowed, more engineering expertise can be incorporated.

Given this discussion, the Naval Undersea Warfare Center/Detachment New London (NUWCDETNLON) and Tecolote Research formulated the following cost engineering model development approach. The goal was to develop engineering-based LCC models for well-defined, narrowly scoped problems. Just as numerous hardware and software components are integrated to build larger systems, many component level LCC models will be used together with engineering inputs to develop system level cost estimates.

The cost model integration approach is based on three steps:

1. The systems engineering and cost estimating teams jointly define components with a limited scope (e.g., fixed, ground-based, C3I software systems),
2. The cost estimating team builds a general purpose cost engineering model that represents the defined component class, and
3. The cost estimate and/or systems engineering teams customize estimate models to allow early design/cost tradeoff analysis. This might involve identifying or building relationships between design parameters (performance characteristics) and cost drivers (equipment specifications such as weight or lines-of-code). This step builds a link between system engineering characteristics and life-cycle costs.

SYSTEM LEVEL COST ENGINEERING

As previously discussed, it is possible to develop robust component level cost engineering models for narrowly scoped problems.

The remainder of this report describes one proposed architecture for integrating cost engineering models into a systems engineering environment. The next section presents a brief discussion of the DESTINATION systems engineering tool set and the Systems Engineering Technology Interface Specification (SETIS). DESTINATION is the systems engineering environment into which a cost engineering capability will be integrated. SETIS provides a standard communications framework for linking the components of the DESTINATION tool set.

DESTINATION AND SETIS DESCRIPTION

DESTINATION

NSWC Detachment White Oak has created an integrated set of systems engineering capabilities under the ECS Research Block. These tools, design aids, methodologies, analysis and optimization technologies, collectively called DESTINATION, provide the capability to perform design synthesis and optimization tradeoffs for large complex systems. The optimization and tradeoffs are performed by allocations of system functions to resources to maximize a set of objectives that the user requires the system to meet. The objectives are represented using the system design factors (SDF) methodology developed by the ECS project.[4] The SDFs are used to define both functional and nonfunctional attributes of the system.

DESTINATION provides an integrated environment for the system engineer that supports design structuring, resource capture, allocation techniques, and optimization strategies. However, the capability to assess cost in a tradeoff analysis has not been implemented within DESTINATION. NUWCDETNLON, which has been assigned the responsibility by NSWC Detachment White Oak to define and incorporate this cost estimation capability, will use the ACEIT framework to support the effort.

SETIS

As described earlier, data transfer between the tools developed by ECS block collaborators for DESTINATION is facilitated by a formal data structure representation called the SETIS. SETIS is a standard interface structure that allows communication between the DESTINATION tools that perform design structuring, resource capture, allocation, and optimization. In order to share data contained in various components of DESTINATION, SETIS provides a common structure in which to extract and transfer data.

Within SETIS, the information used to characterize a system is contained in a system model comprised of three sub models. Together the three models provide a comprehensive system description; alone each model contains the information that represents one view of the system. Thus, the separate models allow the specific information describing a system to be partitioned.

The system model is partitioned as a hierarchy consisting of the logical, implementation, and allocation models. The logical model is used to represent the sequence and interaction of system functions without regard to implementation. This model is configured as a diagram that contains nodes and edges. The nodes represent processes and the edges represent communication between processes. The data contained in the implementation model are partitioned between three diagrams, which represent the software, hardware, and organizational implementation of the system. These diagrams consist of nodes and edges. The final component of the system model, the allocation model, contains information that identifies how specific attributes are allocated to hardware, software, and humanware. In addition, the allocation model contains information that identifies constraints on various aspects of the components of the system.

SETIS supports communication between tools within DESTINATION via formatted ASCII files. The models, diagrams, and attributes used to provide a system description are defined by formatted data structures contained within the files. These ASCII files provide a standard representation of the information used to describe a system as well as the communication medium.

An overview of the ACEIT system is presented next. ACEIT is an automated framework that allows the cost analyst to easily structure LCC models and estimates.

AUTOMATED COST ESTIMATING INTEGRATED TOOLS (ACEIT)

ACEIT is an integrated family of products designed to assist cost analysts in such activities as cost estimating, what-if studies, cost proposal preparation and evaluation, risk and uncertainty analysis, and CER development. It provides a highly automated environment for building detailed cost estimates for complex weapon systems. Although ACEIT is not a cost model, it provides an ideal platform for developing cost engineering component models because it is easily integrated with existing cost models and methodologies.

The main components of ACEIT are

- ACE—a structured spreadsheet style system for preparing cost estimates,
- RI$K—a structured spreadsheet style system for preparing cost estimate risk assessments, and
- CO$TAT—a full-featured statistical analysis system, tailored for the cost analyst.

ACEIT also provides an estimating methodology library, which is a database of catalogued and documented CERs and commercial/noncommercial models available to the ACEIT user. This library provides access to several hundred CERs and models.

ACE—Automated Cost Estimator

ACE is the centerpiece of the ACEIT cost analysis system. Because it is specifically developed by cost analysts, this special-purpose spreadsheet style tool has two basic characteristics that distinguish it from other spreadsheet tools. First, it is organized and structured to follow the steps used in developing a cost estimate, ranging from technical baseline and system definition through time phasing and documentation. Second, the primary tools and techniques of cost analysis have been fully automated in ACE so that the analyst is free to devote more effort to the substance of the estimate rather than to the mechanics. The following list illustrates the principal features of ACE and the high level of automation that has been achieved.

- A user-friendly, intuitive interface that eliminates the need for time-consuming studies of extensive documentation.
- Spreadsheet-style commands and pull-down menus and full-feature on-line help system.
- Built-in WBSs, CESs, and definitions that can be expanded and edited to suit the estimate at hand.
- Built-in cost estimating methodology libraries containing hundreds of CERs, factors, and models readily accessible for use in an estimate. Choices from libraries are brought directly into the estimate along with appropriate descriptive documentation.
- Automated checking of variables, variable names, mathematical expressions, and missing values.
- Automated summation according to WBS/CES indenture levels.
- Automatic normalization for different monetary units, fiscal year, and burden rates; all-inclusive DoD and NASA inflation indices; support for user-defined inflation indices.
- Automated support for all forms and methods of learning theory, including rate effects, shared learning (e.g., multiple users, sites, or platforms), and broken learning.
- Extensive list of predefined functions that can be pasted directly into an estimating equation. Complex functions are available for specialized inflation adjustment, yearly phasing, matrix manipulation, and economic analysis. Many more functions are also accessible.
- Many automated features to support different time phasing methodologies.
- Documentation capability suitable for Blue Book purposes and Cost Analysis Improvement Group (CAIG) review; built-in on-the-fly documentation capability.
- Open architecture. Can easily share data with other ACEIT components. Several commercial cost models include capabilities to export data to ACE. Several spreadsheet-based cost models can export data to ACE. ACE can also export data and results to many other applications, including spreadsheets and databases.

CO$TAT—STATISTICAL ANALYSIS FOR COST ANALYSTS

The CO$TAT system is a PC-based statistics package designed expressly with the cost analyst in mind. As such, it includes only those methods and techniques used in the day-to-day work of cost analysis and does not carry all the statistical methods typically included in standard commercial packages. Not carrying this "overhead" allows CO$TAT to be efficient in its use of memory and fast in its execution. The following list summarizes its automated features and statistical capabilities:

- User-friendly, intuitive user interface.
- Easy data manipulation and normalization; access to ACEIT inflation factor database.
- On-the-fly documentation capability.
- Univariate analysis—measures of central tendency, dispersion, and shape; confidence interval for sample mean; histograms.
- Multivariate regression analysis—linear, log-linear, and nonlinear regression.
- Learning curve analysis—cumulative average theory; unit cost theory; weighted and unweighted least squares.

RI$K—COST ESTIMATE RISK ASSESSMENT

The RI$K model is a general purpose tool that can be used to assess the risk inherent in a cost estimate. The underlying methodology of the RI$K model is based on a thorough understanding of the cost estimation process.

The following list summarizes the automated features and risk assessment capability provided in the model:

- User-friendly, intuitive user interface.
- Several methods for quantifying cost estimating uncertainty. Quantitative measures include prediction interval and low/high ranges. Qualitative measures, which have been calibrated to limited historical data and expert opinion, are also available.
- Qualitative measures for quantifying schedule and technology uncertainty impacts on cost.
- Specification of interrelationships between uncertainties. Factor methods can be used to estimate cost impact uncertainties from one WBS element to another. Generalized approach for specifying complex interrelationships.
- State-of-the-art Monte Carlo solution method as well as an experimental closed-form analytic solution method.
- Wide array of standard reports and graphs.
- Sharing of data with other applications. Can import point estimates from ACE or other commercial spreadsheets. Can export risk results back to ACE or to spreadsheets.

The next section presents a preliminary specification for an interface between ACEIT and DESTINATION by means of SETIS. The interface is illustrated with two examples.

PRELIMINARY ACEIT/SETIS INTERFACE SPECIFICATION

This section first presents a specification for the candidate interface implementation and then illustrates the essential features of the implementation on two sample cost estimating problems: one for software, the other for hardware.

PRELIMINARY SPECIFICATION

It should first be noted that ACEIT is a PC DOS-based application and that SETIS and DESTINATION are UNIX based. While there are some plans to "rehost" ACEIT to Microsoft Windows and X-Windows under UNIX, ACEIT is currently not available as a "built-in" application on these platforms. However, as discussed previously, ACEIT creates a highly specialized environment specifically designed to help cost estimators build life-cycle cost estimates and/or estimating models. It has not been configured for the needs of the system engineer whose primary interest is the end product developed by the ACEIT user, namely the results obtained from the estimating model. In ACEIT, these estimating models are referred to as ACE sessions.

The main components of an ACE session include three ASCII files and two binary files. The ASCII files contain the estimating structure (CES/WBS), all the estimating methodologies and throughputs, and all the input variables that make up the estimate. The binary files contain textual data that describe each cost element and provide a rationale for each methodology and input.

Given the discussion above, it can be seen that the main emphasis should be on developing a generic interface between DESTINATION/SETIS and ACE (namely any cost estimating capability developed during ACE sessions). While it would be possible to rehost ACEIT to UNIX, this is not necessary since the system engineer simply needs to execute ACE sessions on different sets of input values. This can be accomplished by providing a library of ACE sessions (component-level cost engineering models), which would be accessible within DESTINATION. When a cost estimate is required by a component of DESTINATION, SETIS would be used to provide new sets of inputs to an ACE session. A UNIX-hosted ACE Executive (calculation engine only) would be available to run the new inputs through the model and generate a new estimate. The new estimate would then be transferred from the ACE Executive to DESTINATION via SETIS.

BUILDING AND MAINTAINING ACE COST ESTIMATING MODELS

The ACEIT system will be used as the standard environment for producing cost engineering models for integration with DESTINATION. While it would be helpful for ACEIT to be available on the UNIX platform, this is not essential because cost engineers can use the current DOS implementation of ACEIT to build the component-level estimating models. In particular, any ACE estimate or model could be used as the basis for a cost engineering model that could be interfaced via SETIS with DESTINATION.

SETIS INTERFACE FILE

ACE sessions are structured around the CES/WBS structure. Specifically, the methodology data are stored as one large table structure where the rows correspond one-to-one with CES/WBS elements, model inputs, or intermediate calculations. The row structure of an ACE session is completely determined by the estimator. Most of the columns of the table are predefined for a specific estimating purpose, such as specifying a learning curve or addressing inflation. Several additional columns are available that can be used for assigning user-definable key words to different rows. Two of these columns would be used to identify which rows are “visible” to SETIS and DESTINATION.

The SETIS interface column would be used to indicate if a row in a session is an INPUT or an OUTPUT to/from SETIS. If a field in the SETIS interface column is blank, the corresponding row would be considered "invisible" to SETIS. The key word INPUT would be used to indicate a value that DESTINATION must provide to obtain a cost estimate from the model. The key word OUTPUT would be used to indicate a value that would be provided to DESTINATION.

The SETIS code column would be used to define unique identifiers to INPUTs and OUTPUTs of the ACE model. These identifiers would be used to pass data between SETIS and the correct rows in the session. The identifier, and the corresponding row descriptor, would be visible to DESTINATION.

Component-level cost engineering models developed in ACE would then be electronically transferred onto the DESTINATION/SETIS UNIX platform and included in the component cost model library.

UNIX-HOSTED ACE EXECUTIVE

A component-level cost engineering model library would be built on the UNIX platform. This library would contain cost estimating models developed as ACE sessions. Information would be passed between DESTINATION and the models via a SETIS file.

The ACE Executive developed for the UNIX platform would essentially be comprised of the ACE calculation engine. The Executive would process cost estimation requests from DESTINATION via SETIS. Since the normal mode of operation would be as a file processor with only a limited user interface, a high level of portability would be achieved. Initially, the Executive would be developed for compatibility with the DOS, Windows, and UNIX environments.

EXAMPLE 1: SOFTWARE REHOSTING MODEL EXAMPLE

This section illustrates the systems engineering/cost estimation interface process in the context of software rehosting. In particular, the software rehosting problem involves importing the existing software system to a new automated data processing (ADP) environment. This might involve upgrading to new ADP equipment (ADPE) within the current manufacturer product line, changing product lines within a manufacturer, or changing the manufacturer. Problem resolution considerations such as performance and timing, resource contention, precision and accuracy, and interface and interplay have an important role in determining the cost of the effort.

Table 1 presents the basic structure of the modeling approach as implemented in ACE. Row 1 estimates the cost of rehosting the estimating software system to a new ADPE, while row 11 estimates the corresponding new development effort. These would be the primary outputs of the model. Rows 17 through 37 are the model inputs but only a small selection has interest for the systems engineer. Table 2 shows a possible assignment of SETIS interface and code key words. Based on this assignment, the SETIS Interface File might be as illustrated in figure 1.

```
SW_REHOST
{ACEMODEL
(description “Software Rehosting Model”)
(inputs (ID input_list))
(outputs (ID output_list))
}

output_list
{SETISList< ACEOUT*>
(list
(ID rhst_cst)
(ID new_cst)
)
}

rhst_cst
{ACEOUT
(description “Rehosting Cost”)
(value 132.0)
(units $K)
(FY 94)
}
new_cst
{ACEOUT
(description “New Dev. Cost (cross-check)”)
(value 2164.5)
(units $K)
(FY 94)
}
input_list
{SETITList < ACEIN*>
(list
(ID kloc)
(ID ncsci)
(ID mfg)
(ID perf_t)
(ID res_c)
(ID prec_acc)
(ID interface)
)
}

kloc
{ACEIN
(description “CSCI Size (K ADA Semicolons)”)
(value 25)
}
ncsci
{ACEIN
(description “Number of Integrating CSCIs”)
(value 2)
}
mfg
{ACEIN
(description “Manufacturer/Product Line Changes (1-3)”)
(value NULL)
}
perf_t
{ACEIN
(description “Performance and Timing”)
(value 1)
}
res_c
{ACEIN
(description “Resource Contention”)
(value NULL)
}
prec_acc
{ACEIN
(description “Precision and Accuracy”)
(value NULL)
}
interface
{ACEIN
(description “Interface and Interplay”)
(value NULL)
}
```

Figure 1. SETIS Interface File for Example 1

Table 1. Software Rehosting Model—Estimating Methodology (Base Year 94 $K)

	WBS/CES Title	UNIQ ID	EQUATION/THROUGHPUT
1	Rehosting Cost		
2	Analysis and Design		RD*LRATE
3	Code and Unit Test		RI*LRATE
4	Integration and Test		RT*LRATE
5			
6	Rehosting Effort (staff months)		
7	Analysis and Design	RD	BASIS*.4*min(1,PRD/100)
8	Code and Unit Test	RI	BASIS*.3*min(1,PRI/100)
9	Integration and Test	RT	BASIS*.3*min(1,PRT/100)
10			
11	New Dev. Cost (crosscheck)		LRATE*BASIS
12	New Dev. Effort (crosscheck)	BASIS	10*KLOC+10*NCSCI
13			
14	Input Variables	*IN_VAR	
15			
16	*** CSCI Descriptors ***		
17	CSCI Size (K ADA Semicolons)	KLOC	10
18	Number of Integrating CSCIs	NCSCI	0
19			
20	Composite Labor Rate ($K/SM)	LRATE	2.5
21			
22	*** Rehost Adjustment Consideration ***		
23			
24	Manufacturer/Product Line Changes (1 - 3)	MFG	1
25			
26	Problem Resolution Considerations	PRC	
27	Performance and Timing		0
28	Resource Contention		0
29	Precision/Accuracy		0
30	Interface and Interplay		0
31			
32	People Considerations	PC	
33	Not planned for in prior builds		0
34	Different teams		0
35	Different contractor		0
36	Time lag in start		0
37	Prior builds not fully tested		0
38			
39	*** Intermediate Calculation		
40			
41	Redesign Percentage	PRD	
42	System Design Impacts	PRD1	matval(@RDF,1,MFG)+matcoltot(4,@PRC,@RDF+1,MFG)
43	People Impacts		MATCOLTOT(5, @PC, @RDF+5, PRDL)
44	! System Design Impact Level	PRDL	dosum(PRD1>FYIVAL(@BINS,FYFIRST+INDEX),1,6)
45	Reimplementation Percentage	PRI	

Table 1. Software Rehosting Model—Estimating Methodology (Base Year 94 $K) (Cont'd)

	WBS/CES Title	UNIQ ID	EQUATION/THROUGHPUT
46	System Design Impacts	PRI1	matval(@RIF,1,MFG)+matcoltot(4, @PRC, @RIF+1, MFG)
47	People Impacts		MATCOLTOT(5, @PC, @RIF+5, PRIL)
48	! System Design Impact Level	PRIL	dosum(PRI1>FYIVAL(@BINS,FYFIRST+INDEX),1,6)
49	Retest Percentage	PRT	
50	System Design Impacts	PRT1	matval(@RTF,1,MFG)+matcoltot(4,@PRC,@RTF+1,MFG)
51	People Impacts		MATCOLTOT(5, @PC, @RTF+5, PRTL)
52	! System Design Impact Level	PRTL	dosum(PRT1>FYIVAL(@BINS,FYFIRST+INDEX),1,6)
53			
54	*** Factor Matrices ***		
55			
56	Redesign Factors	RDF	
57	Minimum		[Input Throughput]
58	Perf/Time		[Input Throughput]
59	Res Cont.		[Input Throughput]
60	Prec./Acc.		[Input Throughput]
61	Interface/Interplay		[Input Throughput]
62	Not Planned for in prior		[Input Throughput]
63	Different team		[Input Throughput]
64	Different contractor		[Input Throughput]
65	Time lag to start		[Input Throughput]
66	Prior build not fully tested		[Input Throughput]
67	Reimplementation Factors	RIF	
68	Minimum		[Input Throughput]
69	Perf/Time		[Input Throughput]
70	Res Cont.		[Input Throughput]
71	Prec./Acc.		[Input Throughput]
72	Interface/Interplay		[Input Throughput]
73	Not Planned for in prior		[Input Throughput]
74	Different team		[Input Throughput]
75	Different contractor		[Input Throughput]
76	Time lag to start		[Input Throughput]
77	Prior build not fully tested		[Input Throughput]
78	Retest Factors	RTF	
79	Minimum		[Input Throughput]
80	Perf/Time		[Input Throughput]
81	Res Cont.		[Input Throughput]
82	Prec./Acc.		[Input Throughput]
83	Interface/Interplay		[Input Throughput]
84	Not Planned for in prior		[Input Throughput]
85	Different team		[Input Throughput]
86	Different contractor		[Input Throughput]
87	Time lag to start		[Input Throughput]
88	Prior build not fully tested		[Input Throughput]
89			
90	People Factor Bins	BINS	[Input Throughput]

Table 2. ESC Software Rehosting Model—All Columns (Base Year 94 $K)

	WBS/CES Title	BASELINE	SETIS Int.	SETIS Code
1	Rehosting Cost	132.0*	OUTPUT	RHST_CST
2	Analysis and Design	8.7*		
3	Code and Unit Test	26.0*		
4	Integration and Test	97.4*		
5				
6	Rehosting Effort (staff months)	17.1*		
7	Analysis and Design	1.1*		
8	Code and Unit Test	3.4*		
9	Integration and Test	12.6*		
10				
11	New Dev. Cost (crosscheck)	2164.5*	OUTPUT	NEW_CST
12	New Dev. Effort (crosscheck)	280.0*		
13				
14	Input Variables			
15				
16	*** CSCI Descriptors ***			
17	CSCI Size (K ADA Semicolons)	25	INPUT	KLOC
18	Number of Integrating CSCIs	5	INPUT	NCSCI
19				
20	Composite Labor Rate ($K/SM)	7.7*		
21				
22	*** Rehost Adjustment Consideration ***			
23				
24	Manufacturer/Product Line Changes (1 - 3)		INPUT	MFG
25				
26	Problem Resolution Considerations	1.0*		
27	Performance and Timing	1	INPUT	PERF_T
28	Resource Contention		INPUT	RES_C
29	Precision/Accuracy		INPUT	PREC_ACC
30	Interface and Interplay		INPUT	INTERFACE
31				
32	People Considerations	0.0*		
33	Not planned for in prior builds			
34	Different teams			
35	Different contractor			
36	Time lag in start			
37	Prior builds not fully tested			
38				
39	*** Intermediate Calculation			
40				
41	Redesign Percentage	1.0*		
42	System Design Impacts	1.0*		
43	People Impacts	0.0*		
44	! System Design Impact Level	1.0*		
45	Reimplementation Percentage	4.0*		
46	System Design Impacts	4.0*		
47	People Impacts	0.0*		
48	! System Design Impact Level	1.0*		
49	Retest Percentage	15.0*		
50	System Design Impacts	15.0*		

Table 2. ESC Software Rehosting Model—All Columns (Base Year 94 $K) (Cont'd)

	WBS/CES Title	BASELINE	SETIS Int.	SETIS Code
51	People Impacts	0.0*		
52	! System Design Impact Level	2.0*		
53				
54	*** Factor Matrices ***			
55				
56	Redesign Factors	0.0*		
57	Minimum			
58	Perf/Time			
59	Res Cont.			
60	Prec./Acc.			
61	Interface/Interplay			
62	Not Planned for in prior			
63	Different team			
64	Different contractor			
65	Time lag to start			
66	Prior build not fully tested			
67	Reimplementation Factors	0.0*		
68	Minimum			
69	Perf/Time			
70	Res Cont.			
71	Prec./Acc.			
72	Interface/Interplay			
73	Not Planned for in prior			
74	Different team			
75	Different contractor			
76	Time lag to start			
77	Prior build not fully tested			
78	Retest Factors	0.0*		
79	Minimum			
80	Perf/Time			
81	Res Cont.			
82	Prec./Acc.			
83	Interface/Interplay			
84	Not Planned for in prior			
85	Different team			
86	Different contractor			
87	Time lag to start			
88	Prior build not fully tested			
89				
90	People Factor Bins			

EXAMPLE 2: PHASED ARRAY ANTENNA MODEL

This section illustrates the systems engineering/cost estimation interface process in the context of a phased array antenna cost engineering component model. Under the direction of Air Force ESC, Tecolote Research, Inc., developed a family of electronics black box estimating models.[5] Two of these models were used to develop the cost engineering model presented here.

Table 3 presents the basic structure of the model implemented in ACE. Rows 6 and 7, the average unit cost, might be the main outputs from the model. The inputs would be expected production quantity as well as design parameters such as aperture, number of shifters, and power requirements. Table 4 shows a possible assignment of SETIS interface and code key words. Based on this assignment, the SETIS interface file might be as illustrated in figure 2.

Table 3. BBEST Phased Array Antenna—Estimating Methodology (Base Year 94 $K)

	WBS/CES Title	UNIQ ID	EQUATION/THROUGHPUT
1	Total Cost		
2	Development Cost	DEVC	75.935 * PAA_U100^.8555 * PROTOQ^.5529
3	Production Cost	PRDC	PAA_U100
4			
5	Avg Unit Cost		
6	Avg. Dev Cost		DEVC/(QTY+PROTOQ)
7	Avg Prod Cost		PRDC/(QTY+PROTOQ)
8			
9	Input Variables	*IN_VAR	
10			
11	Prod. QTY	QTY	50
12	Prior QTY	PQ	PROTOQ
13			
14	Prod Learning Slope	SLP	90
15			
16	* BBEST Model		
17	* Box Development		
18	Prototype QTY	PROTOQ	2
19			
20	* Phsd Array Ant		
21	100th Unit Cost	PAA_U100	79.607*NSHFT^.24*(TPOW/APER)^.862*.063^TP
22	Aperture	APER	4000
23	Number of Shifters	NSHFT	2000
24	Total Power	TPOW	10000
25	Type (1=tube,0=ss)	TP	1

```
BBEST_PAA
{ACEMODEL
(description "BBEST Phased Array Ant.")
(inputs (ID input_list))
(outputs (ID output_list))
}

output_list
{SETISList< ACEOUT*>
(list
(ID totcst)
(ID dev_ucst)
(ID prd_ucst)
)
}

totcst
{ACEOUT
(description "Total Cost")
(value 12083.0)
(units $K)
(FY 94)
}

dev_ucst
{ACEOUT
(description "Avg. Dev.  Cost")
(value 113.7)
(units $K)
(FY 94)
}

prd_ucst
{ACEOUT
(description "Avg. Prod  Cost")
(value 118.7)
(units $K)
(FY 94)
}

input_list
{SETITList < ACEIN*>
(list
(ID prd_qty)
(ID dev_qty)
(ID aper)
(ID nshft)
(ID tpow)
(ID pa_type)
)
}

prd_qty
{ACEIN
(description "Prod. QTY")
(value 50.0)
}
dev_qty
{ACEIN
(description "Prototype QTY")
(value 2.0)
}
aper
{ACEIN
(description "Aperture")
(value 4000.0)
}
nshft
{ACEIN
(description "Number of Shifters")
(value 2000.0)
}
tpow
{ACEIN
(description "Total Power")
(value 10000.0)
}
pa_type
{ACEIN
(description "Type (1 = tube, 0 = SS)")
(value 1.0)
}
```

Figure 2. SETIS Interface File for Example 2

Table 4. BBEST Phased Array Antenna—All Columns (Base Year 94 $K)

	WBS/CES Title	BASELINE	SETIS Int.	SETIS Code
1	Total Cost	12083.0*	OUTPUT	TOTCST
2	Development Cost	5913.1*		
3	Production Cost	6170.0*		
4				
5	Avg Unit Cost	232.4*		
6	Avg. Dev Cost	113.7*	OUTPUT	DEV_UCST
7	Avg Prod Cost	118.7*	OUTPUT	PRD_UCST
8				
9	Input Variables			
10				
11	Prod. QTY	50.0*	INPUT	PRD_QTY
12	Prior QTY	2.0*		
13				
14	Prod Learning Slope	90.0*		
15				
16	* BBEST Model			
17	* Box Development			
18	Prototype QTY	2.0*	INPUT	DEV_QTY
19				
20	* Phsd Array Ant			
21	100th Unit Cost	68.5*		
22	Aperture	4000.0*	INPUT	APER
23	Number of Shifters	2000.0*	INPUT	NSHFT
24	Total Power	10000.0*	INPUT	TPOW
25	Type (1=tube,0=ss)	1.0*	INPUT	PA_TYPE

CONCLUSIONS

This report presents a preliminary architecture for integrating cost estimating capabilities into a systems engineering environment. The ACEIT system is used to implement component-level cost engineering models. A library of component models would be available to the systems engineer via the SETIS communication system. System level cost engineering models would be built up from the component level models in the library in much the same way as a complex weapon system is built up from a collection of lower level hardware and software subsystems.

The proposed interface provides an integrated framework that will encourage design engineers and cost analysts to communicate early in the system life cycle. Cost models will be responsive to engineering inputs and design engineers will be able to utilize the expertise of the cost analysts. This level of communication will increase our ability to develop truly cost effective systems.

REFERENCES

1. T. C. Choinski and D. J. Organ, "Integrating Cost Models With Systems Engineering Tools," *Proceedings From the Complex Systems Engineering Synthesis and Assessment Technology Workshop*, Washington, DC, 25 July 1994.

2. The Department of Defense, *Draft Military Standard for Systems Engineering (MIL-STD-499B)*, 6 May 1992, p. 8.

3. E. Rechtin, *Systems Architecting: Creating and Building Complex Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1991, p. 4.

4. C. M. Nguyen and S. Howell, *System Design Factors: The Essential Ingredients of System Design*, NSWC Technical Report 92-268, Naval Surface Warfare Center, Division Dahlgren, Dahlgren, VA, May 1992.

5. "The Black Box Estimator (BBEST) Electronics Cost Models," TT-0011/1, Tecolote Research, Inc., vols. I and II, Santa Barbara, CA, June 1988.

INITIAL DISTRIBUTION LIST

Addressee	No. of Copies
Defense Technical Information Center	12
Advanced Research Projects Agency (Dr. J Muñoz)	1
Office of Naval Research (Dr. D. Moran, E. Wald)	2
Program Executive Office/Submarines (Capt. D. Burgess, Capt. Sabatini Cdr. Anderson)	3
Program Executive Office/Undersea Warfare/ASTO (Cdr. Polcari, W. Chen G. Kamalakis, R. Zarnich)	4
Program Executive Office/Submarine Combat and Weapon Systems (S. Lose, R. Campbell)	2
Program Executive Office/PMO-425 (Capt. Kent, M. Basilica, N. Cook, W. Johnson, C. Pelverts)	5
Space and Naval Warfare Systems Command/PMW-182 (Capt. B. Gallemore, Cdr. S. Hollis, Cdr. E. Reinke, J. P. Feuillet)	4
Naval Surface Warfare Center, Dahlgren (S. Howell)	1
Naval Surface Warfare Center, White Oak (C. Nguyen)	1
Naval Research Laboratory (D. J. Kaplan, R. Hillson)	2
Tecolote Research, Inc., California (J. J. McGahan, A. Sjovold)	12
Tecolote Research, Inc., Massachusetts (R. W. Thompson)	6
Ascent Logic Corp., Massachusetts (M. Morgal)	1
Ascent Logic Corp., Pennsylvania (C. Schuster)	1