

## **ACEIT 6.0: AN ESTIMATE INTEGRATOR**

By John McGahan, Alfred Smith  
Tecolote Research Inc.

### **ABSTRACT**

The Automated Cost Estimating Integrated Tools (ACEIT) is a government funded tool set that has been developed over the past 15 years for and by military cost estimators. Many organizations use several cost estimating tools to build their overall cost model. For over 5 years, ACEIT users have exploited the ACE Executive to run what-if analysis on ACE models from within Excel or (if they did not have ACE) via their browser from Knowledge Web Services (KnWS) sites.

Over 220 sites worldwide rely on ACEIT to build their Life Cycle Cost (LCC) models. Most, if not all of these organizations use ACEIT in conjunction with other cost estimating and engineering tools. Generally, ACEIT is used to “bring together” the analysis from a variety of sources into one cohesive, complete and flexible cost model. This paper will demonstrate how the Automated Cost Estimator (ACE) has been modified in ACEIT 6.0 (scheduled release in the fall of 2003) to establish direct links into other parametric cost estimating models and other COTS systems. We will demonstrate dynamic interfaces between an ACE-based cost model and the PRICE-H model as well as the SEER-SEM model. In addition we will demonstrate a dynamic interface between ACE and Microsoft Project to illustrate the benefits to linking cost and schedule information.

The software (called “Plug-Ins”) for each of these interfaces will be shipped with ACEIT 6.0 and is immediately available to be added to the ACE menu structure by the user. Additionally, the protocol necessary for any developer to build their own “Plug-In” is available upon request.

## Overview of the ACEIT Suite

Integration between software systems can take on many different guises. One of the most common forms of integration is simply to embed key functionality from one application into another. From the end-user view, there is only a single interface, that of the primary application. For all practical purposes, the embedded application acts as a silent partner working quietly in the background to support the primary application. In other situations the separate applications are more loosely coupled. Neither application is primary. Instead, two or more tools are merged into a super-suite that provides limited exchange of data via a central repository. Often times, a custom integration framework is provided to allow ad-hoc calculations, data manipulation, and reporting. In many of these situations, spreadsheets are exploited to establish the integration because they are almost universally used throughout the target user community and they have strengths in data access, calculation, and reporting.

The need for integration of cost estimating models into the tools of other domain areas has been recognized for many years. Initiatives such as Design-To-Cost, Cost As Independent Variable, and recent work in standards for the modeling and simulation community document some of the approaches taken. Furthermore, there is also a need to integrate cost estimating models and tools amongst themselves, as well as with engineering tools. In this paper, we will discuss several techniques used to integrate applications with the Automated Cost Estimator (ACE) based cost models. Many have already been built and have been in use for many years. Once built they can quickly be adapted from one application (e.g., Ship Construction Cost Engineering) to the next (e.g., Aircraft Cost Engineering). In this paper we are introducing the latest techniques, namely direct integration of another cost modeling application into ACE to allow the user to “embed” the other application directly into the ACE model.

ACEIT is a revolutionary, automated architecture and framework for cost estimating and other analysis tasks, developed by and for analysts. ACEIT has several integrated tools that provide many of the critical functions used throughout the cost community. The major components of ACEIT include:

- **ACDB** Warehouses and retrieves raw and normalized cost and technical data
- **COSTAT** Designed specifically for cost estimating statistics and regression analysis
- **AIM** Develop estimating methodology libraries
- **ACE** Automate inflation, learning, phasing, risk, documentation and other essential cost estimating processes to help build a robust, accurate and defensible cost model
- **POST** Program Office Support Tool runs an ACE session from Excel, provides superior case management and auto-generated reports/charts.

In addition, ACEIT includes several other utilities to help estimate system installations across many sites, create and share inflation databases, etc. Knowledge Web Services (KnWS) is also available to establish secure, web-based knowledge and model repositories as well as cost estimating (CERs) libraries that the user can copy from and paste to an ACE session.

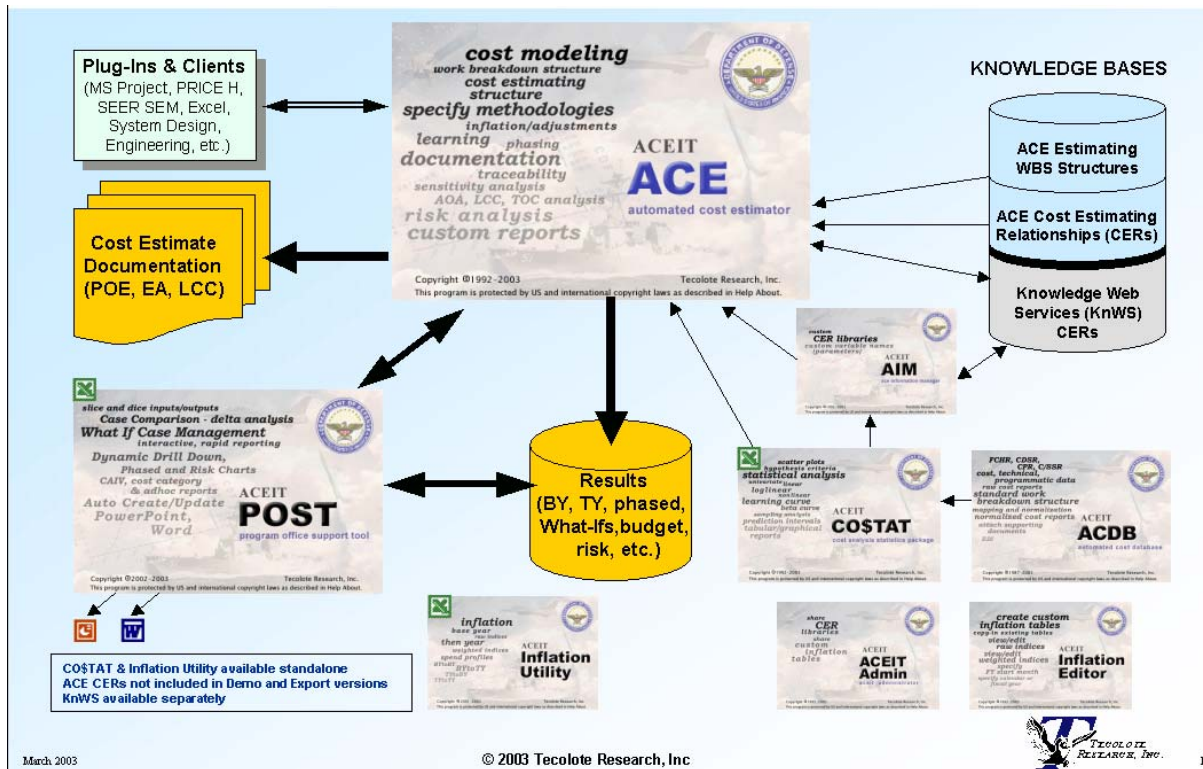


Figure 1: ACEIT Architecture

Automated Cost Estimator (ACE) is the cost estimating portion and heart of the ACEIT platform. It is a model-building, estimate reporting tool. Part spreadsheet, part database, it incorporates enough structure to eliminate most of the automation tasks you would have to develop manually if you started with a spreadsheet, and ACEIT is flexible enough to support nearly any kind of cost analysis. Analyst productivity is increased through built-in indenture summing, inflation, learning, time phasing, documentation, sensitivity/what-if, error trapping, risk analysis and, other analysis capabilities. COSTAT is designed to perform the most common cost estimating statistical analyses efficiently within Excel and pass the results directly into an ACE session. ACE will place the COSTAT data or equation in the correct location within the cost model and include the relevant documentation. ACE also provides the ability to conduct objective (parameters set from a COSTAT analysis) or subjective (user defined inputs) risk analysis within the ACE session. Risk assumptions (distribution shape, bounds, schedule/technical penalties and correlation) can be applied to CER inputs (configuration risk) or at the CER level (cost risk) or both.

From a typical end-user (rather than model builder) perspective, the ACE Executive is an Excel-based interface that allows users to access ACE models to perform what-if drills, perform risk studies, generate risk-adjusted budget reports, and generate ad-hoc charts and tables. However, the ACE Executive is really a two-part application. The primary workhorse of the Executive is the ACE calculation server module. It provides the interface that allows other applications to communicate with ACE cost models. ACEIT also provides a generic Excel-based client application (developed in Visual Basic) that interacts with the ACE calculation server to render

ACE model functionality within Excel. This Excel-based client application is often referred to as the ACE Executive.

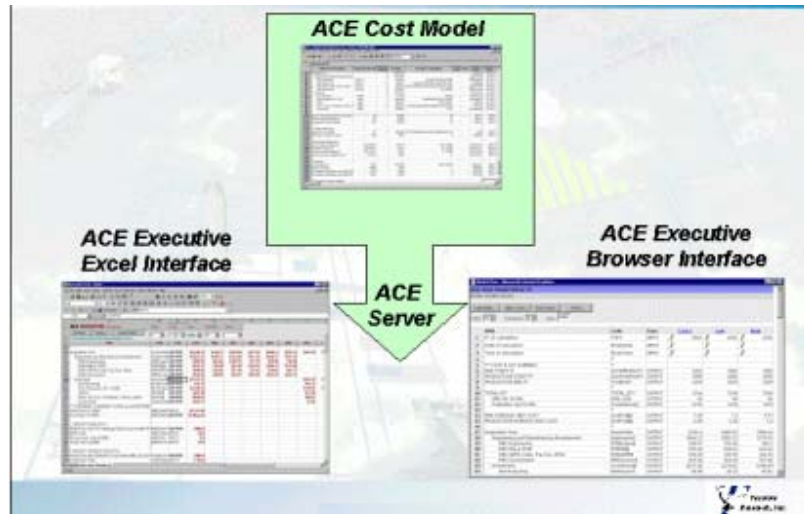


Figure 2: ACE Executive

POST is a new client interface that automates many typical program office functions and will be included in the ACEIT 6.0 release. The POST client uses the same calculation server as the original Executive client. Eventually, POST will take the place of the ACE Executive. It provides the analyst unprecedented views of the cost estimate inputs and results. It will also contain a large number of standard reports and charts that can be customized by the user. Program office staff will immediately see the huge advantage of POST once they realize how easy it will be to pass tailored reports and charts into PowerPoint or WORD and update them automatically.

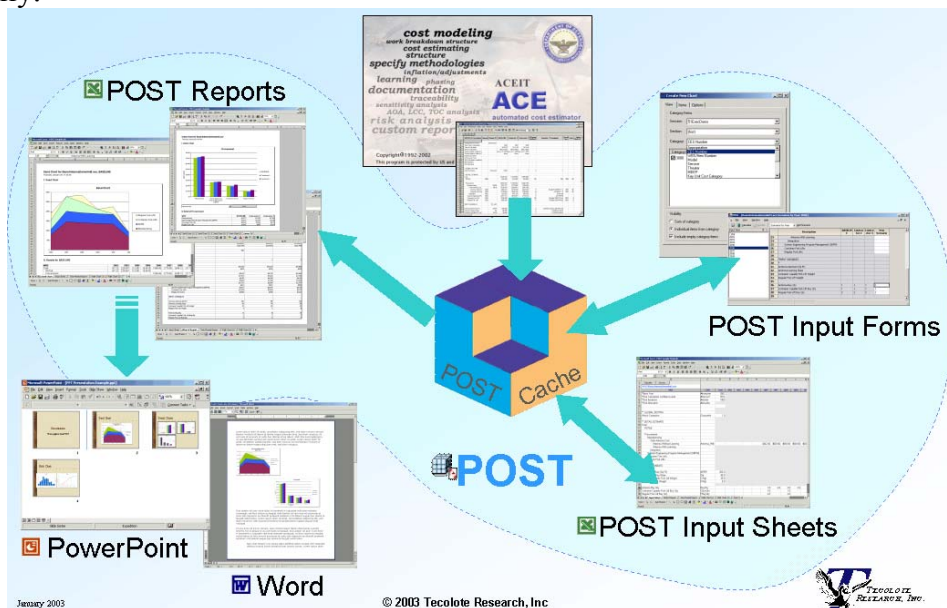


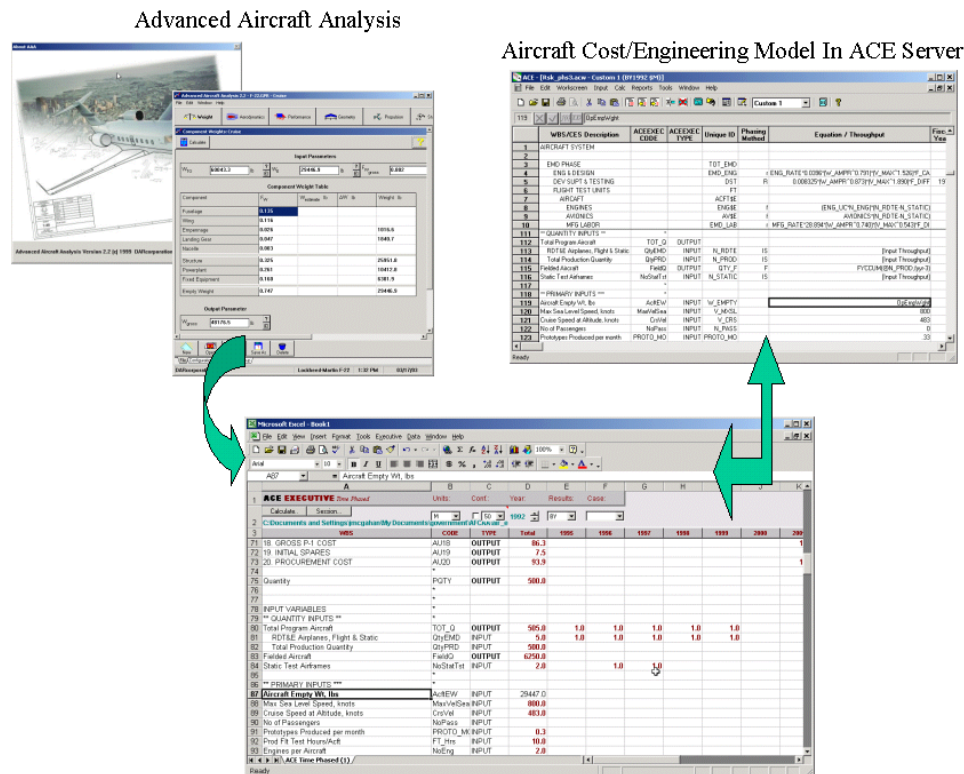
Figure 3: Program Office Support Tool

## **The Evolution of Integrating Other Applications with ACEIT**

The direction and degree of application integration is a function of the end-user needs. Applications from different domain areas (e.g., engineering vs. cost) can often be tightly integrated together when the dataflow requirements are very well defined, the tools and practices are sufficiently evolved to support integration, and there is a strong need (reduce cycle time, improve accuracy) for improving dataflow between domains (e.g., program managers can obtain rapid cost estimate feedback for alternative engineering inputs if the data transfer is automated). With this type of interface, one domain will often have a dominant view. For example, a cost model may be embedded within the engineering tool. Loose integration between different domain tools is often used to facilitate communication between user communities without providing a dominant view. Such tools are usually intended as Integrated Product Team (IPT) support frameworks that will be used by both user types.

In 1994 under an effort sponsored by Naval Undersea Warfare Center (NUWC/NLD), Tecolote Research, Inc, began to develop a general-purpose framework for integrating ACE cost models with engineering tools. The purpose of this effort was to link a family of cost models to a government-developed design synthesis tool, DESTINATION, for specific complex computer systems requirements. The DESTINATION system allowed the design engineer to identify low-level hardware, software, and humanware components that comprise the system and ensure various system requirements are met (e.g., power, heat dissipation, reliability, bandwidth, etc.). ACEIT was used to host a family of component-level life cycle cost models for each of the low level components which could be used to define the complex computer system (e.g., Power Supply, Display, Processor). A UNIX-based ACE model server (ACE Executive) was developed to allow other applications (DESTINATION) to execute these cost models. As a system design was developed from components within DESTINATION and the components specified, appropriate cost models were executed. In this way, ACE and the Executive acted as the framework for hosting cost models that are embedded within another application. During this same time frame, the Naval Surface Warfare Center, Carderock Division, (NSWC/CD) initiated an effort to integrate a sealift ship construction life cycle cost model with a government-developed ship design tool: Advanced Surface Ship Evaluation Tool (ASSET). In this application, ACEIT was used as the platform for hosting the cost model, and the ACE Executive was used to link the cost model to ASSET. In this project, the government modified ASSET to generate an ASCII file in a format that the ACE Executive could read. The Excel client let the analyst load design scenarios from ASSET containing hundreds of cost drivers that characterized a complete, balanced ship design. The analyst was free to combine these with other required cost model drivers, and develop a life cycle cost estimate for sealift ship alternatives. Standard reports and graphs required by NAVSEA were created within the Excel client and linked to the ACE cost model outputs and engineering data from ASSET. The graphs provided instant insight into where there were inconsistencies in the source data or the cost model design. They also helped the analyst identify potential program areas where cost savings could be achieved. Versions of this application have been in constant use in several NAVSEA new ship projects.

A similar technique was used by the Air Force Cost Analysis Agency (AFCAA) to link a commercial aircraft-engineering tool with a supersonic fighter cost model hosted in ACE. The DAR Corporation's Advanced Aircraft Analysis (AAA) tool was selected for this effort. It provides modules to perform aircraft weight sizing, analysis of flight dynamics, stability control, as well as mission profiling and links to a CAD system. For this integration effort, a library of aircraft project files was developed in AAA that could then be used as a starting point for trade studies and analysis of derivative aircraft. For example, the F-22 project file might be used as the starting point for an analysis of a JSF-like system. Again, the Excel client application was used to combine parameters from AAA with other key cost model inputs to support rapid development of trade studies.



**Figure 4: Aircraft Cost Engineering Model**

The process of using ACE and the ACE Executive (with Excel client) to link cost models with engineering systems provides a powerful, general purpose framework for implementing Cost As an Independent Variable (CAIV) analysis. For life cycle cost to be a consideration early in the life of a system, cost models must be sensitive to design issues. The engineering design applications define the parameters that form the basis of the technical trade studies. Furthermore, the engineering systems provide a potential link between costing and system performance. In this way, budget considerations can be filtered back through the design to identify a system solution that yields the best performance given the available funding. By driving the cost model backwards, target technical system descriptors (e.g., weight, power, speed, range, etc.) can be set. These technical targets become inputs into the engineering process that then ensures design feasibility and provides an assessment of resulting system performance.

More recently, there has been great interest in developing links between distributed systems. In a demonstration effort sponsored by Air Force Research Laboratories (AFRL), an ACE-based radar cost engineering model was linked with a radar performance simulation tool via a distributed web-link to form a virtual model, RADEVAL. The users interface was provided via a browser. The radar performance tool was run from a remote Unix server and the ACE-based cost model from a third site. For this example, the radar performance simulation tool AIRADE (Airborne Radar Detection), developed by Technology Service Corporation (TSC), allowed the user to generate radar range performance data for varying radar antenna mainbeam gain and transmitter peak power. This performance data was then passed into the radar life cycle cost model, REACT (Radar Engineering and Cost Tool), developed by Tecolote. Simulation and cost model results were then displayed back to the browser.

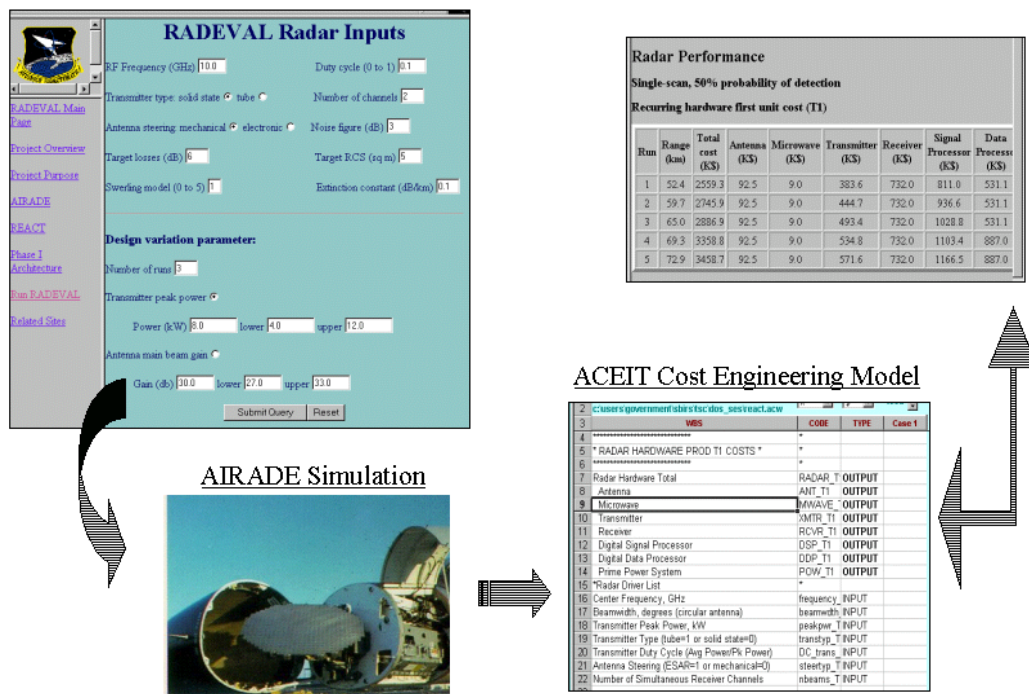


Figure 5: Radar Performance/Cost Evaluation Model

The integration of cost functionality from ACE with other systems can be accomplished in several ways. The simplest technique, using the standard Excel-based ACE Executive client interface or the newer POST client that comes with ACEIT, requires minimal to no programming skills and no knowledge of the Application Programming Interface (API) contained within ACE. Even if you are limited to the most rudimentary programming capabilities contained within Excel, it is possible to develop highly sophisticated interfaces and reporting functions. As many third-party tools contain native Excel interfaces, this is usually the ideal jumping off point for most requirements. This approach was used for the interfaces for ASSET and AAA. For more involved interfaces such as the web-based RADEVAL example, it is necessary to use the ACE Executive Server API together with a modern programming language (such as Visual Basic or Visual C++).

## **Use ACE As The Estimate Integrator**

All of the examples discussed so far have used ACE as a server to feed costs into other applications. These arrangements rely on other applications to provide cost model inputs that the ACE model uses to generate the cost results for use within the decision making process. But, it is also possible to have ACE serve as the dominant view in an integrated tool suite. For example, we might be interested in providing an integrated framework of cost estimating and analysis support tools with a single user view, namely ACE. For this type of interface, the primary function is to transfer ACE data out to and extract information from multiple sources support the cost estimate. To meet this requirement, ACE would need to provide integrated access to common cost estimating tools, models and data sources used in the cost community. Ideally, the user would have some control over this “two way” transfer of data.

ACE provides two approaches for integrating information from other applications into ACE. The simplest technique requires minimal programming skills and does not rely on the ACE API. This technique uses the “smart clipboard” format to push data into specific cells within the ACE framework. The most common application of this technique usually involved creating a custom export format in the information source/model in order to export data in an ACE-readable format. Most commonly an application will include such commands like “Export to ACE” on the File menu. This might export one or more rows worth of data to the ACE clipboard, push equations into the ACE “methodology” column, and paste text into the ACE “definition field” to be integrated into a basis of estimate report. Clipboard interfaces are typically fast, reliable and easy to develop, making them the ideal starting point for initial development. This technique has been used for interfaces with many tools, including both government tools as well as commercial parametric models (PRICE and SEER products had this feature for many years). This technique is used within the ACEIT suit, for passing data from ACDB into COSTAT and transferring COSTAT regression results, risk ranges, and documentation to ACE, AIM or KnWS (web-based AIM libraries). A more complex example of this type of interface can be seen with the ACEIT System-by-Site Wizard utility. Via a sequence of dialogs and forms, the user is able to specify system procurement schedules for multiple sites, identify component configurations per site, and define unit costs and learning curves. When the specifications are complete, the data is exported to the ACE clipboard allowing the user to identify where the data will be “pasted” into ACE.



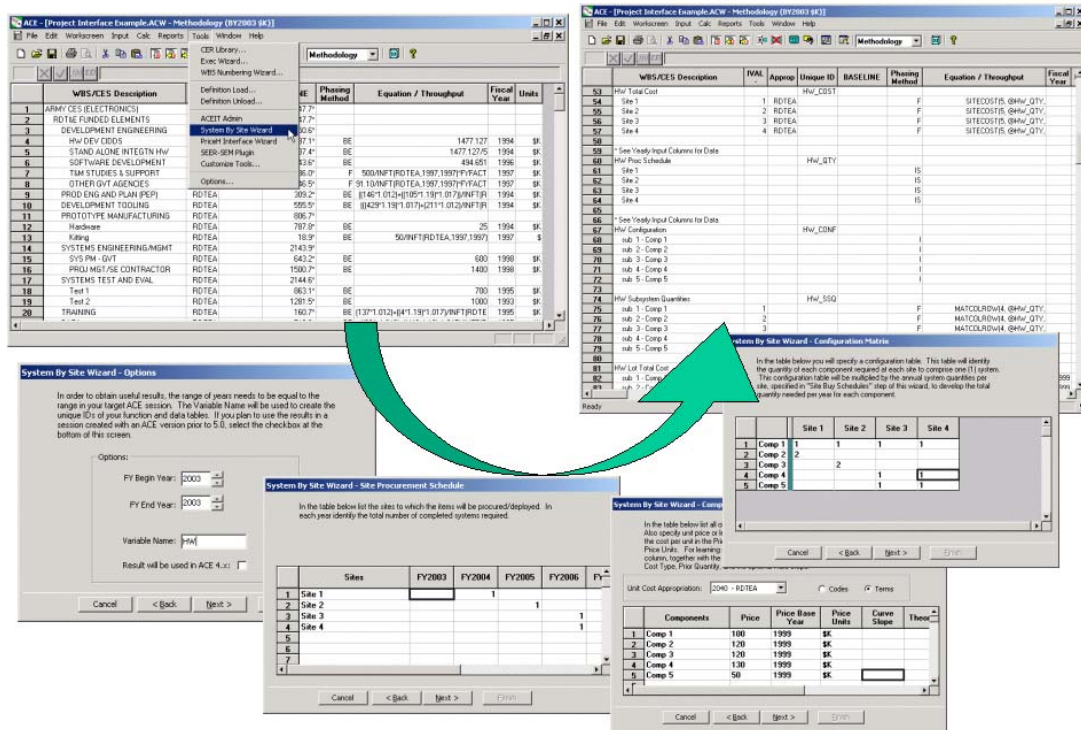


Figure 6: System By Site Wizard

The primary limitation of these clipboard interfaces is that they only work in one direction. Once the information is integrated into ACE it becomes disconnected from the original data source. Microsoft's embedding and linking technology could be used to address this concern but this brings more problems to the table than solved (we have found the technology to be slow, inefficient, and unstable). A more reliable approach exploits, again, the ACE API (as well as the third party application API, usually available for modern Windows applications). The ACE API is available to current ACE sites upon request.

In addition to the ability to embed ACE model functionality within other applications, the ACE API also provides functionality to ***embed other applications within ACE*** using a "plug-in" approach. A plug-in acts as middleware between ACE and an associated third-party application. Plug-ins are applications or components that interact with ACE via the API as well as the third-party application (hopefully via its API). They can be as simple as a data translation routine mapping data between applications with minimal to no user interaction or complex interfaces performing complicated data manipulations complete with all-encompassing user interactions. ACE plug-ins are installed using the "Customize" option under the ACE Tools menu. Tecolote created several plug-ins using Visual Basic. VB is a powerful development language that provides many functions specifically designed to facilitate communication with Common Object Model (COM) automation APIs. This makes VB an ideal language for rapid development of middleware utilities.

## **Plug-Ins Shipped with ACEIT**

We will now review three plug-ins that interact with the following third-party applications:

- PRICE-H Interface
- SEER-SEM Interface
- Microsoft Project Interface

These interfaces all make use of the Dynamic Equation Column (DEC) capability in ACE. This feature allows users to add new equation columns to their ACE session, each with a unique identifier. DECs are created by “type” consistent with the data type that they will contain: numbers, costs, dates, or comments. DECs are a convenient location to store the information that will be exchanged between ACE and the target applications via its plug-in. For example, with the MS Project interface, DECs are used to exchange project start dates, end dates, and the activity durations. In addition, DECs are used to “remember” the link to the source application file so that the plug-in can be used again to refresh the exchanged information.

### **PRICE-H Interface Example**

PRICE-H is the parametric hardware cost model in the PRICE suite of models. The PRICE suite also includes a hardware life cycle model (HL), an electronics and microcircuit model (M), and a software model (S). The need for an interface between ACE and PRICE-H arose from a requirement of the European Space Agency (ESA). They utilize ACE to develop early-on acquisition cost estimates for their programs. During this process they utilize data from several sources including an organic cost/price database system, methodologies libraries, as well as commercial parametric models, the most common being PRICE-H. Different estimating techniques are used to estimate different elements of the Work Breakdown Structure (WBS). For selected rows, they may wish to use PRICE-H.

For this application, ESA chose to implement a small family of PRICE-H template files calibrated to ESA experience for equipment classes of interest. Within their ACE-modeling framework, they then created the DECs required to contain critical design parameters (e.g., weight, complexity, etc) that would drive the linked PRICE H model. Additional DECS are created to store the associated PRICE-H outputs (e.g. Total Cost). For any cost element, ESA is then able to make changes to any PRICE H input assumptions and obtain results in a very compact and convenient way. The results are then fully available to the ACE session for further analysis, phasing, inflation, learning, combining with other costs, risk analysis or any other cost analysis function.

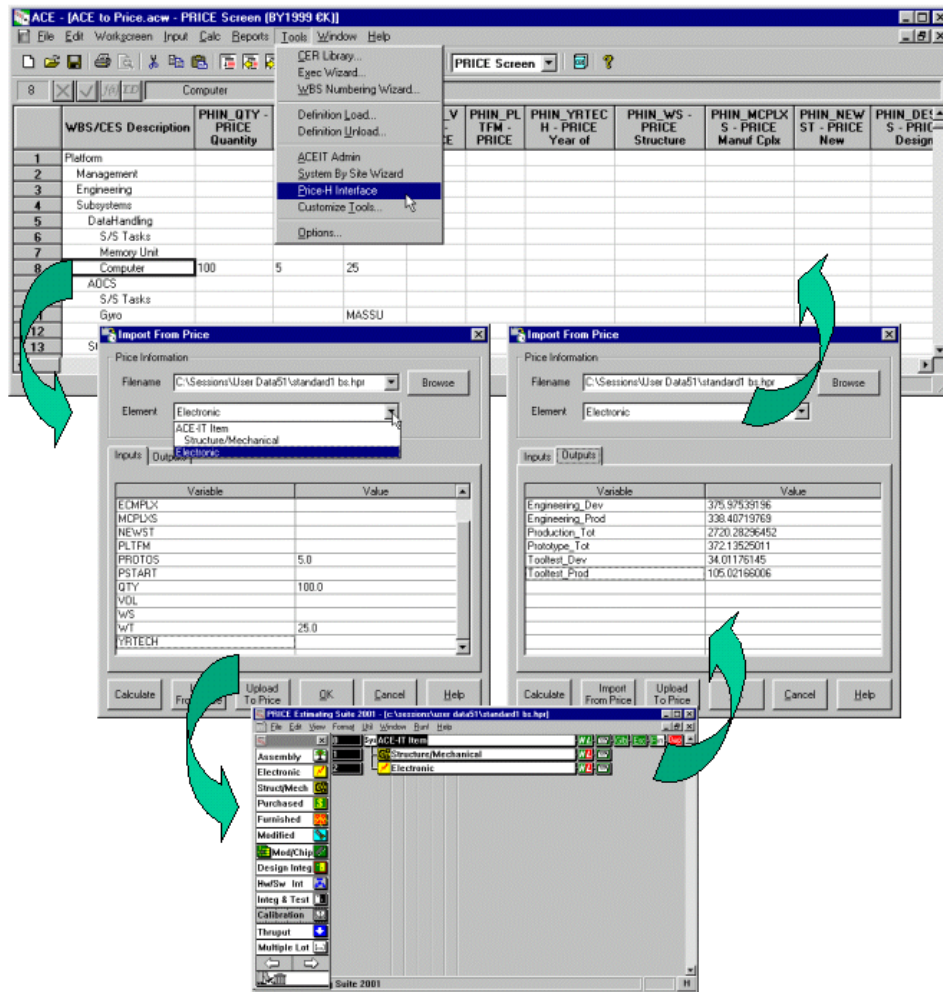


Figure 7: PRICE-H Plug-in Interface

## SEER-SEM Interface Example

SEER-SEM is the software cost model in the SEER family (which also includes a hardware model, and a design for manufacturing model). The basic design, look and operation for this plug-in is similar the PRICE-H plug-in. A SEER-SEM template file was developed to act as the data exchange interface between ACE and SEER-SEM. DEC's are added into the ACE estimate to allow key SEER SEM input parameters to be exchanged (e.g., software size, complexity, effort adjustment, parameters, estimated effort, cost, etc). The unique aspect of this interface is the lack of a true automation API in the third-party tool. In this case, SEER-SEM did not provide a COM-based API but rather, a more involved server mode interface. The interface plug-in has to negotiate between the ACE API and the SEER-SEM server model interface. There is little difference to the end user between these interfaces but the underlying technology, and software development issues, was significantly different. Nevertheless, once the data exchange process was defined, SEER SEM is as easy to integrate into ACE as any other application.

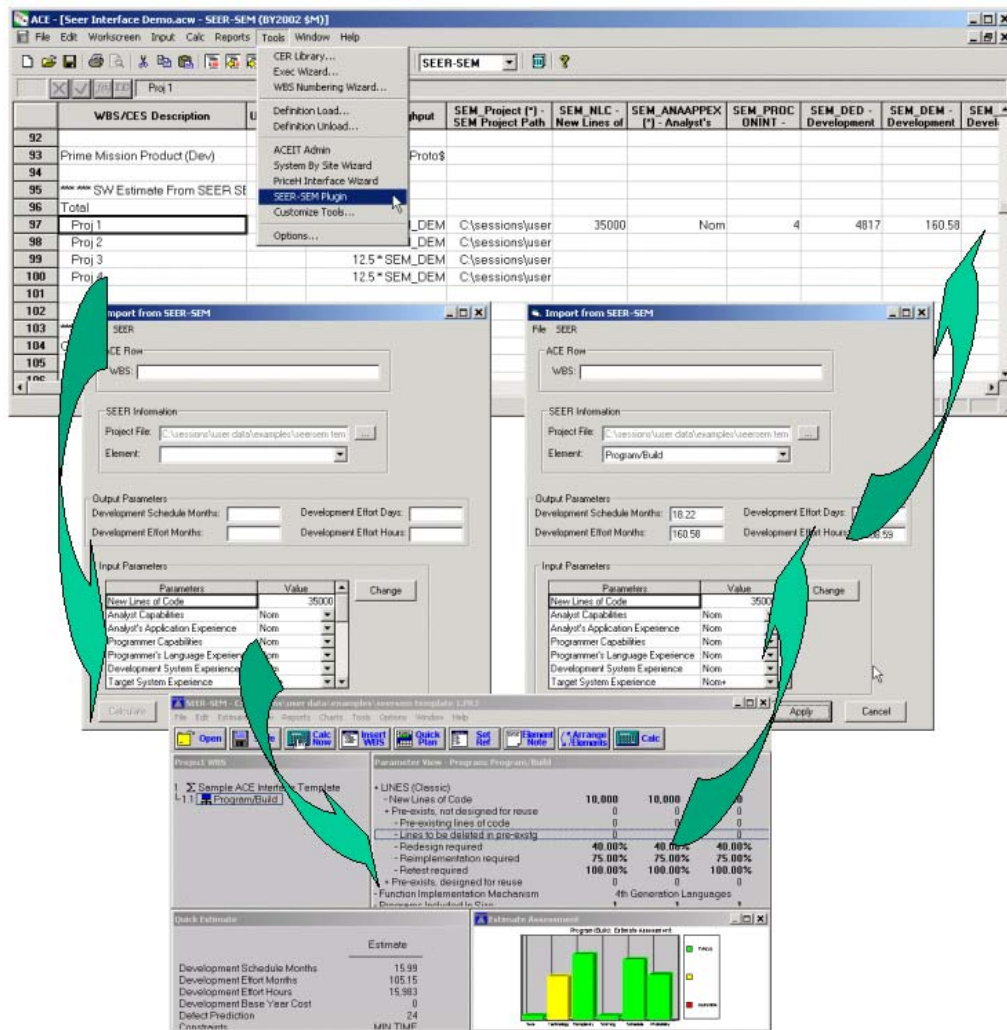


Figure 8: SEER-SEM Plug-in Interface

## Microsoft Project Interface Example

The Project interface uses many of the same interface concepts as the SEER and PRICE interfaces but with a very different goal in mind. With the SEER and PRICE interfaces, the primary goal was to allow ACE users to easily access the different parametric models to estimate specific elements of their WBS, using PRICE on some rows, SEER on other, and home-grown parametric methods, learning curves, factors, and analogies on others. Each application of the interface is treated as a separate row within the estimate structure.

The Project interface is intended to support schedule/cost what-if drills within a single interface. A typical ACE-based model will estimate costs phased over time to allow reporting of budget (escalated) funds. As the development phase of a program is typically incrementally funded, development estimates will tend to be highly schedule sensitive. More often than not, the phasing assumptions used for the purposed of estimating costs in these situations can quickly diverge from the schedule as defined in the ever changing master program schedule. The Project

plug-in addresses this problem by allowing the user to define links between the master program schedule in MS Project and the appropriate time phasing assumptions in the cost estimate.

As with the other plug-ins, DEC's are used to define Project interface parameters that will be exchanged with ACE, including the Project file name and path, activity codes, start date, end, date, and duration. In the case of a resource loaded schedule, it would also be desirable to exchange cost information. Unlike the other plug-ins, this interface is essentially one-way. The primary flow of data is from Project into ACE (keeping the cost estimate phasing in sync with the master schedule). The plug-in approach is used as a means of keeping the information up-to-date rather than as a bi-directional exchange. Further enhancements to pass resource cost data back into Project is possible, but to date there has not been a requirement to do so.

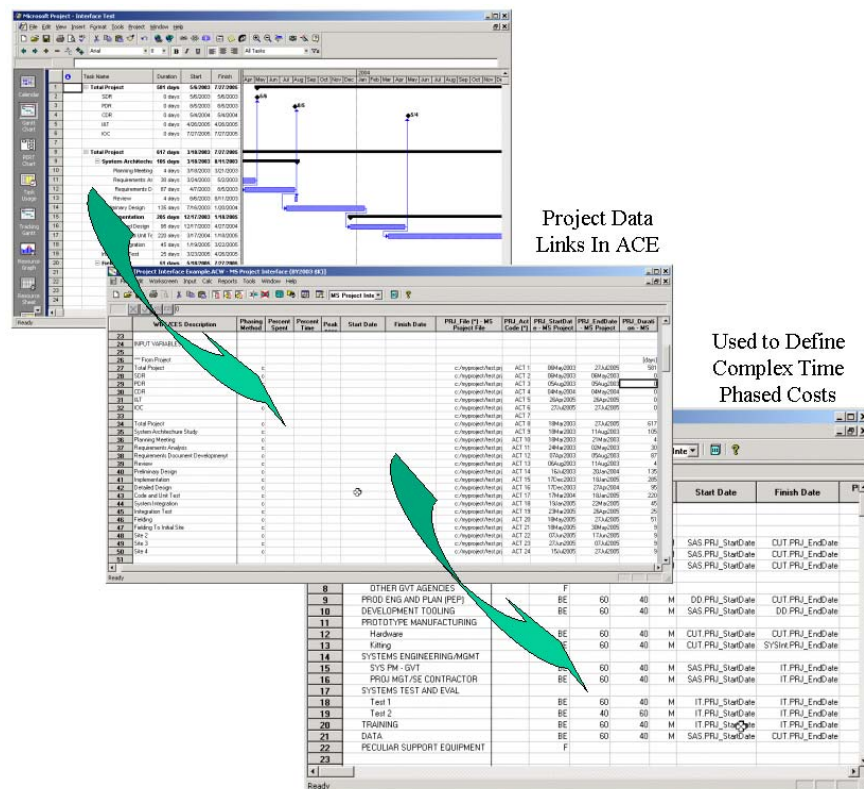


Figure 9: Microsoft Project Plug-in Interface

## Conclusion

The use of Application Programming Interfaces (API) allows applications to exchange information and, if desired, allows applications to be embedded within one another. To achieve quick, reliable results, clipboard exchanges and even simple data file handshakes are often sufficient. It is often desirable to use a spreadsheet framework as the integrator when custom user interfaces, complex data manipulations and specialized reporting are necessary. Spreadsheets provide a rich development framework together with powerful data manipulation, charting, and reporting functions.

We have demonstrated that the Automated Cost Estimator can perform as an overall Estimator Integrator, easily interoperating with other ACEIT components for parametric model and learning curve development (CO\$TAT), stored libraries of estimating methods (AIM, KnWS), as well as wizard add-ins such as the System-By-Site Wizard. Utilizing the new plug-in approach and DEC's, it is also possible to quick interchange data with third-party parametric models (SEER, PRICE) as well as commercial tools (Project). These interfaces greatly enhance the users ability to develop rapid, substantiated cost estimates within a single integrated framework.

## **Speaker Biography**

**John McGahan**: ACEIT Product Development Manager, senior software systems analyst, cost analyst, and statistician. Mr. McGahan joined Tecolote Research Inc. in 1984 and has over 15 years of cost analysis, estimating, and model development experience. He is the project leader for all ACEIT development efforts, the focal point and lead designer for the ACEIT RISK model, and the chief scientist for the Tecolote Software Products and Services Group. Education: 1991 University of Texas at Austin, Graduate Studies (Operations Research); 1984 Dartmouth College, Hanover, New Hampshire M.A. (Mathematics); 1982 Fredonia State University, Fredonia, New York B.S. (Mathematics/Computer Science).

### **Address:**

John McGahan  
Tecolote Research, Inc.  
5266 Hollister Ave. No 301  
Santa Barbara, CA 93111

**Tel:** (253) 756-1670

**Fax:** (253) 752-1527

**E-mail:** [jmcgahan@tecolote.com](mailto:jmcgahan@tecolote.com)

**Alfred Smith**: General Manager for the Software Products/Services Group. Over 20 years experience leading, executing or contributing to life cycle cost model development for wide variety systems such as Aircraft Carriers, Submarines, Multi Mission Aircraft, Amphibious Assault Vehicles, International Space Station and Customs Modernization. 10 years experience teaching cost estimating, statistical analysis and risk modeling. Tecolote lead for the development, distribution and support of the ACEIT suite of tools. Education: 1986 University College, London, England, Master of Science with Distinction; 1978 Royal Military College, Kingston, Bachelor of Engineering.

### **Address:**

Alfred B. Smith  
Tecolote Research, Inc.  
5266 Hollister Ave. No 301  
Santa Barbara, CA 93111

**Tel:** (805) 964-6963

**Fax:** (805) 964-7329

**E-mail:** [asmith@tecolote.com](mailto:asmith@tecolote.com)