System of Systems Cost Estimating Solutions

Josh Wilson

PRICE Systems, LLC

System of systems is a relatively new term being applied to capability driven projects addressing large, inter-disciplinary problems involving many systems at multiple levels and multiple domains. Being poorly understood overall, it is not surprising that the tools and thought processes needed to address such problems are incomplete. The need to solve system of systems problems is urgent not only because of the growing complexity of modern challenges, but also because system of systems problems involve decisions that commit large amounts of money and resources. Fortunately, an area in which several advocates are attempting to improve industry understanding of system of systems relates to estimating the cost and effort required to deliver capabilities rather than platforms. Currently, the significant offerings in this and closely related realms include COSYSMO, COSOSIMO, SECOST, SEER-H with TSV, PRICE PES, and PRICE TruePlanning Systems. This paper explores the history of each of these models, the origin of their methodologies and cost estimating relationships, the major size and cost drivers, and the extent to which each solution addresses the need for a system of systems estimating capability. But first, what exactly *is* a system of systems?

Maier's Criteria for defining a system of systems has probably been encountered by anyone spending at least a few minutes looking into the subject. There are a few additional characteristics that often appear, but for the most part a system of systems may be classified as such by exhibiting a majority of the five traits listed by Maier: Operational Independence, Managerial Independence, Evolutionary Development, Emergent Behavior, and Geographical Distribution. Another way to define a system of systems is by citing the ways in which system of systems engineering (SoSE) differs from traditional systems engineering. According to the System of Systems Engineering Center of Excellence, the modern transition to capabilities based processes has introduced challenges that traditional systems engineering was never intended to address. While a systems engineer is concerned with developing to a specific, well-defined requirement in a stable architecture, SoSE works in a dynamic architectural environment to build a broadly defined capability that enables interoperability. Unique behaviors beyond those exhibited by any single component are avoided in systems engineering because they lead to instability, but in SoSE such emergent behaviors are leveraged to enhance performance, flexibility, and adaptability. The differences between systems and systems of systems truly are fundamental and not superficial. Despite the existence of these many differences, the first attempts to estimate the costs associated with creating a system of systems still sought to exploit proven systems engineering cost models, such as COSYSMO.

A part of the COCOMO family of cost models from the University of Southern California, COSYSMO was developed as a research project intended to help people reason about the economic implications of systems engineering on projects (Valerdi, 2006). The first version of COSYSMO was created by a Raytheon affiliate in 2001. It was called "MyCOSYSMO" and was based on the aforementioned research taking place at the Center for Software Engineering (now the Center for Software and Systems Engineering, or CSSE) at USC. These efforts at USC culminated in the authoring of a doctoral dissertation by Ricardo Valerdi in August 2005. Shortly thereafter, academicCOSYSMO was made available by USC. This single sheet Excel implementation of COSYSMO provided an estimate of systems engineering effort based on the size and cost drivers defined by Dr. Valerdi in his dissertation. Based on feedback from the users of academicCOSYSMO, a 28 page document covering COSYSMO scope, usage, output, and more was written, and the academicCOSYSMO User Manual was available in July 2006. Currently, efforts are being made to improve COSYSMO through the incorporation of reused and other non-new requirement types, risk, and labor and schedule spreading capabilities. Efforts are also being made to obtain more data and to map out a process for organizations, potentially even those with little training or parameter knowledge, to successfully deploy COSYSMO.

The system engineering activities used by COSYSMO are based on the ANSI/EIA 632 *Processes for Engineering a System* standard. At a high level, the standard covers five fundamental processes: Acquisition and Supply, Technical Management, System Design, Product Realization, and Technical Evaluation. These processes are broken into thirteen process categories, which are then divided even further into 33 low-level activities. Technical Management, for example, is broken into Planning, Assessment, and Control processes, and the Planning process includes activities such as Process Implementation Strategy, Technical Effort Definition, and Work Directives.

Life-cycle phases in COSYSMO are based on ISO/IEC 15288 - *System Lifecycle Processes*. According to this standard, the six ordered life-cycle phases of a system are Conceptualize, Develop, Operational Test and Evaluation, Transition to Operation, Operate, Maintain, or Enhance, and finally Replace or Dismantle. Only the first four of these phases, however, are within the scope of COSYSMO, because for the last two (Operate, Maintain, or Enhance and Replace or Dismantle) the information provided by affiliates for the model yielded too little data to calibrate (Valerdi, 2005).

The source of data for COSYSMO is thirty-four projects from six companies in the aerospace and defense sector. Raytheon, BAE, General Dynamics, the Aerospace Corporation, Northrop Grumman, and Lockheed Martin provided data, and three of these companies were responsible for twenty-seven of the thirty-four data sets upon which COSYSMO is based. One of the strengths of COSYSMO is that it is open-source and data driven, but a possible drawback is the narrow field of organizations from which the data was obtained, of which anyone using the model should be aware.

The outcome of the USC research based on this data was model with a set of four size drivers and fourteen cost drivers that provides a point estimate of systems engineering effort in person months. The four COSYSMO size drivers are System Requirements, Interfaces, Algorithms, and Operational Scenarios. Each of these size drivers is converted to an equivalent number of Nominal New System Requirements and summed to get the size of the system. Fourteen cost drivers then determine the effort required to engineer a system of the calculated size. The fourteen cost drivers used by COSYSMO are Requirements Understanding, Architecture Understanding, Service Requirement Level, Migration Complexity, Number and Diversity of Installation Platforms, Recursive Levels, Documentation, Technology Risk, Stakeholder Team Cohesion, Team Capability, Team Experience, Process Capability, Multisite Coordination, and Tool Support (Valerdi, 2005).

The end result of the COSYSMO effort is a parametric model intended to estimate person months of systems engineering as defined by the International Council on Systems Engineering (INCOSE). INCOSE defines systems engineering as "an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem." As stated earlier, there are many differences between systems and system of systems engineering. The INCOSE definition for a system of systems is "an interoperating collection of component systems that produce results unachievable by the individual systems alone." These many differences are why COSYSMO is a less than ideal option for estimating system of systems, and probably also why USC decided to continue its system and system of systems engineering research to develop COSOSIMO.

The potential need for a system of systems engineering model was identified by USC and others in early 2003. By late 2003 an initial COSOSIMO (which stands for the Constructive System of Systems Integration Model) had been developed based on software size, and in the fall of 2004 the early design was revised based on SOS architecture (Lane, 2006). Throughout 2005 and 2006 several analyses and surveys were conducted at SoSE conferences and workshops, and the COSOSIMO model morphed into a 2-submodel and then a 3-submodel version, the idea being that each submodel will have fewer parameters more tailored to associated SoSE activities. Recent efforts have also been made to incorporate COSOSIMO into the Enterprise Systems Engineering (ESE) process model for SoSE and associated Enterprise Architecture Management Framework (EAMF) developed by Dr. Paul Carlock of Northrop Grumman Missile Systems and Robert Fenton. This process model and framework include a set of processes and activities devoted to capability delivery design and integration throughout an enterprise's mission planning (Lane & Carlock, 2006).

Like COSYSMO, COSOSIMO is based on research from USC, and is also a part of the COCOMO family of cost models. It is being developed primarily by Ph.D. candidate Jo Ann Lane, and the goal is for fall 2007 availability. COSOSIMO will estimate the SoSE costs at the system of systems level and will not include development costs for any subsystems. Specifically, the characteristics of systems of systems supported by COSOSIMO include having strategically oriented stakeholders interested in tradeoffs and costs, having a long-range architectural vision for the SOS, the existence of a lead systems integrator (LSI) responsible for developing and integrating the SOS, and system component independence (Lane, 2006).

As mentioned earlier, the current version of COSOSIMO takes the form of three submodels, each covering different activity areas. Each of these submodels has its own set of size and cost drivers, although some size and cost drivers are used in more than one of the submodels. The Planning, Requirements Management, and Architecting submodel has as its size drivers the Number of SOS Related Requirements and the Number of SOS Interface Protocols. The submodel for Source Selection and Supplier Oversight has only the Number of Independent Component System Organizations as a size driver. The SOS Integration and Testing submodel has three size drivers: Number of SOS Interface Protocols, Number of SOS User Scenarios, and Number of Unique Component Systems. For each submodel, the size drivers are used to calculate total size, and the cost drivers adjust the amount of effort required to architect and deliver a system of systems of the calculated size. The cost drivers for COSOSIMO are widely shared throughout all three of the submodels, and include Requirements Understanding, Architecture Maturity, Level of Service Requirements, Stakeholder Team Cohesion, SOS Team Capability, Maturity of LSI Processes, Tool Support, Cost/Schedule Compatibility, SOS Risk Resolution, Component System Maturity and Stability, and Component System Readiness.

COSOSIMO is being designed and developed specifically for estimating SoSE costs, meaning the results from using the model will not include SOS development costs or any subsystem development or systems engineering costs. Pending sufficient participation in current and ongoing SoSE surveys and data contributions from both SOS and systems engineering programs, COSOSIMO will be available in the fall of 2007.

Another COSYSMO-based offering for systems and SoSE estimating is Raytheon's SECOST. As mentioned previously, the first version of COSYSMO was built by a Raytheon affiliate in 2001. This initial "MyCOSYSMO" model was leveraged off of a model called SWCOST developed at Raytheon's Intelligence and Information Systems (IIS) Garland, TX location. SWCOST had been used successfully at Garland and other IIS sites to estimate systems engineering costs for over eight years (Ilseng, 2006). Once COSYSMO was near completion at USC and MyCOSYSMO had been sufficiently circulated to gain confidence, a proprietary version of MyCOSYSMO was developed at Raytheon. This model, called SECOST, has been deployed at several Raytheon business units to be used as a second opinion for proposals.

Functionally, SECOST is a suite of several Excel spreadsheets. It has a fifteen step cost estimation process that includes Document Project Assumptions, Document and Register Project Risks, Set COSYSMO Size and Cost Drivers, Determine Labor Distributions Among Raytheon Salary Grades, Time Phase the Systems Engineering Estimate, Submit to Pricing Group, and Conduct Internal Estimate Review (Ilseng, 2006). Since it uses COSYSMO as the embedded engine, its activities are also based on the EIA/ANSI 632 *Processes for Engineering a System* standard, and its schedule is based on ISO/EIC 15288 *System Lifecycle Processes*. Basically, SECOST is COSYSMO tailored specifically to Raytheon in an Excel framework. SECOST interfaces with standard Raytheon Pricing Systems, the labor for various activities is distributed across Raytheon Salary Grades, and capability to time phase an estimate has been built in.

SECOST does not account for many SOS characteristics and is therefore not an ideal tool to estimate SoSE effort, but with certain Raytheon recommended modifications and additions to the COSYSMO size and cost drivers presented by Jon K. Ilseng at the 21st International Forum on Systems, Software, and COCOMO Cost Modeling in Herndon, Virginia, SECOST can be used for SoSE estimation. To account for the often extensive software level requirements found in many SOS, additional size drivers for Software Requirements and Software Modules are needed. Besides these two new size drivers, a modification to the existing System Interfaces Complexity size driver is recommended to capture the complexities involved in managing SOS interfaces. New cost drivers are also required, including an IV&V factor to rate the maturity and experience of the team in performing IV&V tasks at the SOS level, and an SOS integration factor to rate the maturity and experience of performing as a LSI. Typical changes to the cost drivers include adding the word "SOS" in the appropriate place and possibly modifying the viewpoints to more appropriately address SOS issues. The cost drivers that require modification are Requirements Understanding, Migration Complexity, Number and Diversity of Installation Platforms, Number of Recursive Levels in the Design, Documentation to Match Lifecycle Needs, Stakeholder Team Cohesion, and Personnel Experience/Continuity (Ilseng, 2006).

So far, each of the models discussed have been derivatives of or extensions to COSYSMO. Galorath's SEER-H with Total System Vision (TSV) is a commercial tool, unrelated to COSYSMO. SEER-H with TSV was developed after completing a study of 45 out of 120 NASA/Air Force Cost Model (NAFCOM) projects along with limited data from other sources. Partially funded by the NASA Independent Program Assessment

Office (IPAO), the goal of the study was to identify system level costs, characterize statistical relationships to other project costs and parameters, and finally develop cost estimating relationships (CERs) to enable total system estimating capability. By October 2005, modifications to the SEER-H cost model had been completed. The initial goal of developing CER based costs, however, was abandoned in favor of using a simple percentage of development cost for development system level costs (SLC), and of first unit production costs to estimate production SLC (Stump, n.d.).

Upon completion, SEER-H with TSV added capability to provide system level cost inputs at any rollup level in the work breakdown structure. Based on percentage of subsystem costs, the production and development SLC costs may be adjusted at each rollup depending on user selections for complexity and experience. Both production and development SLC costs are subdivided into subsets of five SLC component activities. These five SLC components, which are based on five of the six cost categories used in NAFCOM, are System Engineering and Integration (SEI), Integration, Assembly, and Test (IAT), System Program Management (SPM), System Test Operations (STO), and System Support Equipment (SSE). SLC costs for development may include all of the above listed categories, but only the first three categories (SEI, IAT, and SPM) were decided to be appropriate for production (Hunt, 2006).

It is unclear whether the capabilities provided by SEER-H with TSV are sufficient to estimate engineering costs at the SOS level. SEER-H with TSV is believed to be more applicable to compact systems such as aircraft, ships, and ground stations than to highly distributed systems (Stump, n.d.). Also, SEER-H does not allow software cost estimation, so all system level costs are therefore based strictly on hardware cost. Integration with software estimates is available, however, through usage of a SEER ADDIN cost element, allowing SEER-SEM elements to be linked directly into a SEER-H with TSV work breakdown structure. In addition to this capability, more than one system level assignment can be made in a given work element structure, meaning that SLC cost rollups may be nested, and each of them may calculate any or all of the five applicable SLCs for development or production.

Another of the commercially available tools that can potentially be used to estimate SoSE costs is the PRICE Estimating Suite (PES) by PRICE Systems. PES includes parametric models to estimate costs associated with hardware, software, and microcircuits. Specifically, the models include a Hardware Acquisition model (introduced in 1975), a Hardware Lifecycle model (1976), a Software Development model (1977), a Software Lifecycle model (1979), and a Microcircuit cost model (1983). In addition to these models, PES includes an Excel Solution addin. PRICE Excel Solution essentially allows any of the models to be run and updated through an interface to Excel.

The basic idea behind the PES SOS solution is that individual component systems can be accurately modeled using any combination of the models listed previously. Integration and Test and Design Integration at the system level can be accounted for by using available PES components, as can Hardware/Software Integration. The PES framework, however, does not allow hardware components and software components to exist in the same file. This can be bypassed through the use of Thruput elements, which act in a similar manner to SEER ADDIN cost elements. Thruput elements allow the cost of completed software estimates to be added to files containing hardware components. Systems and system of systems costs beyond that calculated by the Integration and Test, Design Integration, and Hardware/Software Integration components provided by the model must be accounted for by additional means, typically at the discretion of the estimator, and likely involving Excel. The existence of PRICE Excel Solution makes this fairly simple, though, since information from a PES file may be pulled into Excel and manipulated as desired. Still, successfully modeling a system of systems requires the estimator to make sound decisions about systems engineering and SoSE costs. Another drawback to this and the other previously discussed solutions is a result of the multiple levels of integration required to model an SOS; due to the fundamental nature of systems and systems of systems, the estimator is inconvenienced by the need to manipulate and combine data from several models.

The final offering available in the realm of system and SOS estimating solutions is also a PRICE Systems tool. PRICE TruePlanning Systems is a catalog of cost objects that operates within the activity-based cost estimating framework provided by PRICE TruePlanner. The True Systems catalog has been developed based on data from the PRICE Systems KnowledgeNetwork and research and analysis from USC, Cranfield University, SEI, the Aerospace Corporation, Defense Acquisition University, and MIT.

True Systems has been designed specifically to enable system and system of systems cost estimating capabilities using existing PRICE TruePlanning catalogs, such as True S for software estimating, True IT for information technology, and True H for hardware. This integrated capability results from the existence in True Systems of parent System and Assembly cost objects whose costs are driven by information from children components in the work breakdown structure. Similar to COSOSIMO's three submodel design, the Assembly and System objects are tailored to different specific SOS-related activities. The Assembly cost object accounts for the cost of technical activities that occur during development of a system, such as defining system requirements, designing the system, and integration and test. Project level oversight and control activities that are not direct component development activities are accounted for by the System component (Minkiewicz & Shermon, 2006). Examples of System activities include project management, quality assurance, and configuration management. The two cost objects may be nested or used alone in any location of the WBS to capture the appropriate activities, presenting a high-fidelity solution to cost estimators. All of the activities for both System and Assembly also have associated resources, which represent anything money is spent on to perform an activity. The resources required to perform the requirements definition activity, for example, are Project Stakeholder, Business Analyst, and Systems Engineer.

The system engineering and SoSE size drivers for both System and Assembly include information from children components in the work breakdown structure, as well as any children Systems or Assemblies. Depending on the type of component (hardware or software) and the child activity type (development, production, or operation and support), different information is passed to the System or Assembly parent. This information may include labor totals for development or production or both, integration sizes, integration complexities, or various combinations of these values. Once the appropriate sizes are determined based on the information rolled up from children, the total labor required to perform each activity is determined from System and Assembly cost drivers. There are also system engineering related inputs such as number of requirements, interfaces, and operational scenarios on the System and Assembly that function as additional size drivers for system engineering activities.

The cost drivers for both the System and Assembly cost objects are user inputs such as Project Complexity, System Complexity, Stakeholder Involvement, Requirement Stability, and Operating Specification. Since the System and Assembly cost objects cover different activity areas, only the appropriate adjustment factors are found as inputs to each cost object. Because of its use of a single framework taking advantage of already existing cost models, its activity based nature, and its division of activity responsibility into two primary cost objects, True Systems offers a self-contained and flexible option for estimating system and SOS costs.

Estimating the cost and effort required to develop and integrate systems and system of systems is a difficult undertaking in a field that is still obviously maturing. Each of the major solutions available to the estimator seeking to capture such costs have been covered: COSYSMO, COSOSIMO, SECOST, SEER-H with TSV, PRICE PES, and PRICE True Systems. The history, methodologies, size and cost drivers, similarities, and differences of these approaches to the system of systems problem have been discussed. Each approach has strengths and shortcomings, and none can reasonably claim to fully portray the complexities or report the efforts needed to build systems of systems with total accuracy. The good news is that the need for such capability is recognized, and effort is being made to research what is available, make progress with our understanding of the problem, and to provide the tools needed to make informed decisions.

	COSYSMO	COSOSIMO	SECOST	SEER-H with TSV	PRICE Estimating Suite System of Systems	PRICE True Systems
Who Made It	USC Center for Software and Systems Engineering Ricardo Valerdi, Ph.D.	USC Center for Software and Systems Engineering Jo Ann Lane, Ph.D. candidate	Raytheon – originally developed at their Intelligence and Information Systems (IIS) Garland, TX location	Galorath, Inc.	PRICE Systems	PRICE Systems
Availability	academicCOSYSMO – single Excel spreadsheet Free	Anticipated Fall 2007 pending data Free	From Raytheon - "used with permission"	Available as Add-In to SEER-H Commercial	Available as PRICE Estimating Suite Commercial	Available as a catalog in TruePlanner Commercial
Interoperability	Standalone systems engineering model	Standalone system of systems engineering model	Standalone systems engineering model	Plugin to SEER-H Interacts dynamically with SEER-SEM files through use of ADDIN elements from another plugin	PRICE H, PRICE S must operate separately	Hardware, Software, IT, and System cost objects interact within single TruePlanning framework
What it Estimates	Systems Engineering effort (person months)	SoS definition and integration effort (person months)	Same as COSYSMO but phased over time and broken into activity buckets	The costs associated with integrating a coherent, functioning system	Acquisition costs for hardware and software components	System - costs of conceptualizing, planning, managing , deploying and maintaining a project Assembly - technical activities that occur during the development of a system
Number of Activities and Resources	None, although guidance is provided in dissertation for division of point estimate into activity and phase buckets	3 activities aligning with submodels No resources	Uses Excel capabilities to break COSYSMO estimate into EIA 632 activities and EIC schedule phases Resources divided by Raytheon Salary Grades	8 activities – all cost categories for development and first three (SEI, IAT, SPM) for production No resource info available	No SoS specific activities or resources Component activities include Development, Production, Support Component resources include Labor and Material	System • 6 activities Assembly • 11 activities 18 resources

	COSYSMO	COSOSIMO	SECOST	SEER-H with TSV	PRICE Estimating Suite System of Systems	PRICE True Systems
Activity Coverage	Based on ANSI EIA 632 Standard • Acquisition & Supply • Technical Mgmt • System Design • Product Realization • Technical Evaluation	Based on 3 Submodels • Planning, Requirements Management, & Architecting • Source Selection & Supplier Oversight • SoS Integration & Testing	Same as COSYSMO	Based on 5 of 6 NAFCOM cost categories • System Engineering & Integration (SEI) • Integration, Assembly, Test (IAT) • System Program Management (SPM) • System Test Operations (STO) • System Support Equipment (SSE)	No built in SoS activities Component activities include development engineering, development manufacturing, production engineering, production manufacturing	System • Project Initiation and Planning • Project Management • Quality Assurance • Vendor Mgmt • Documentation Assembly • Requirements Def • System Design • Development Eng • Development Man • Development TnT • Production Eng • Production Man • Production TnT • Software Integration & Test • System Integration & Test • Operational Test & Evaluation
Schedule Coverage	ISO EIC 15288 subset • Conceptualize • Develop • Operational Test and Eval • Transition to Operation	• Inception • Elaboration • Construction • Transition	Same as COSYSMO	No info available	No built-in SoS schedule Component schedule Acquisition through Deployment	• Conceptualize • Plan • Manage • Deploy • Maintain
Major System or System of System Drivers	• # Requirements • # Interfaces • # Algorithms • # Operational Scenarios • 14 Team & Application Adjustment Factors	• # SoS Requirements • # SoS Interface Protocols • # SoS User Scenarios • # Unique component systems • # Component system organizations • 11 Adjustment Factors	Same as COSYSMO • # Requirements • # Interfaces • # Algorithms • # Operational Scenarios • 14 Team & Application Adjustment Factors	• Hardware Development Cost • First Unit Production Cost • Team Experience • System Complexity	At the discretion of the estimator (Excel or other tool needed to estimate SoS level costs)	• HW/SW/IT Development Labor • HW First Unit Production Cost • SW/IT Production Labor • HW/SW/IT Operation and Support Labor • # Requirements • # Interfaces • # Vendors • # Operational Scenarios • Requirements Stability • Project/System complexity
Sources of Data	34 projects from 6 Aero & Defense companies	No info available	Same as COSYSMO	45 of 120 NAFCOM projects w/ limited data from other sources	PRICE Knowledge Network	PRICE Knowledge Network, Research and analysis from USC, Cranfield University, SEI, Aerospace Corporation, Defense Acquisition University, MIT

References

Callaway, R., DeLaurentis, D., Bankes, S.C., & Popper, S.W. (2004). System of Systems Symposium: Report on a Summer Conversation.

Hunt, Robert P. (2006). "Capturing Total System of Systems Costs Using SEER-H with TSV and SEER-SEM". Proceedings from the 21st International Forum on Systems, Software, and COCOMO Cost Modeling. Herndon, VA.

Ilseng, Jon K. (2006). "Systems Engineering Cost Estimation: System of Systems". Proceedings from the 21st International Forum on Systems, Software, and COCOMO Cost Modeling. Herndon, VA.

Lane, Jo Ann. (2006). "Constructive System of Systems Integration Cost Model (COSOSIMO) Tutorial". Proceedings from the 21st International Forum on Systems, Software, and COCOMO Cost Modeling. Herndon, VA.

Lane, Jo Ann, & Carlock, Paul. (2006). "System of System Enterprise System Engineering, the Enterprise Architecture Management Framework, and System of Systems Cost Estimation". Proceedings from the 21st International Forum on Systems, Software, and COCOMO Cost Modeling. Herndon, VA.

Lane, Jo Ann, & Carlock, Paul. (2006). "COSOSIMO: How Well Does it Capture the Unique Aspects of System of Systems Engineering Processes?". Proceedings from the 21st International Forum on Systems, Software, and COCOMO Cost Modeling. Herndon, VA.

Maier, Mark W. (1998). Architecting Principles for Systems-of-Systems. *Systems Engineering, Volume 1* (Issue 4), 267-284.

Minkiewicz, Arlene, & Shermon, Dale. (2006). "System of Systems Workshop". Proceedings from the 26th Annual PRICE Systems European Symposium. Prague, Belgium.

Stump, Evin. (n.d.) "Estimating All Costs at System Level." Retrieved January 4, 2007, from http://www.galorath.com/pdf/TechTipEstimatingCostsSystemLevel.pdf.

Stump, Evin. (n.d.) "Estimating System Level Costs." Retrieved January 4, 2007, from http://www.galorath.com/presentations/Estimating_System_Level_Costs.pdf.

Valerdi, Ricardo. (2005). The Constructive Systems Engineering Cost Model (COSYSMO). (Doctoral dissertation, University of Southern California, 2005).

Valerdi, Ricardo. (2006). academicCOSYSMO User Manual: A Practical Guide for Industry and Government.