Paper #167

On the Use of Architectural Products for Cost Estimation

Ricardo Valerdi
Massachusetts Institute of Technology
Cambridge, MA
rvalerdi@mit.edu

Indrajeet Dixit
University of Southern California
Los Angeles, CA
idixit@usc.edu

Abstract

The Department of Defense Architecture Framework (DoDAF) provides a standard set of views that illustrate specific attributes of a system. These views give different levels of detail and purpose that allow engineers to express operational, system, technical, and architectural properties for specific purposes. The twenty six different views available can be useful and at the same time overwhelming to someone unfamiliar with the framework.

An increasing number of defense contractors are using DoDAF to characterize system attributes. These same contractors are responsible for providing cost estimates for the development and implementation of systems. This paper provides the link between these two areas by relating architectural views to system representation for cost estimation. There are several benefits to this link. First, the cost estimation community can benefit from a deeper understanding of the DoDAF and its objectives to improve the field of cost estimation through the development of models that better represent system architectures. Second, DoDAF can serve as a common language between customers and contractors by improving the representation of stakeholder needs and objectives. Third, the architecting community can benefit from the identification of subjective cost drivers currently not addressed in the DoDAF products.

In this spirit, this paper describes how DoDAF architecture frameworks can be used to determine functional system size for adequate estimating of systems engineering effort. This is illustrated through the use of the OilCo FastPass system defined in previous work. The utility of using the FastPass system is that it is well documented in journal articles and it is a system familiar to the general systems engineering audience.

Background

Systems Engineering Cost Estimation. As organizations develop more complex systems, increased emphasis is being placed on Systems Engineering (SE) to ensure that cost and schedule are within budget. Correspondingly, the failure to adequately plan and fund the systems engineering effort appears to have contributed to a number of cost overruns and schedule slips, especially in the development of complex ground and space systems. Government and commercial organizations have recently placed increased emphasis on accurately planning the SE function and on understanding the factors that influence the resources needed to implement and perform SE.

In an attempt to better quantify the SE activity, the DoD acquisition community has been exploring *Systems Engineering Revitalization* and migrating back to a reinstatement of prescribed military standards. Several models and tools have become available to aid in forecasting systems engineering resource needs, but few guidelines exist to help engineers and program managers determine which approach is best suited for estimating any particular effort.

It became apparent that the systems engineering community had an opportunity to heavily leverage the work done for software cost estimation by the University of Southern California Center for Software Engineering (USC-CSE.) The CSE had the research methodology, knowledge, and tools based upon their recent COCOMO II development project to develop an extension to address systems engineering cost. COSYSMO, the Constructive Systems Engineering Cost Model, is an "open" model that has been developed by consensus of the Corporate Affiliates for the purposes of estimating systems engineering costs.

To date, the cost modeling community has not capitalized on the DoD Architecture Framework (DoDAF) which was explicitly developed to characterize multiple layers of system attributes. To highlight the benefits of DoDAF in the cost estimation environment, an example system is used to illustrate the architecture views relevant for cost modeling. Artifacts for this system include use case diagrams, DoDAF views, and an enterprise architecture model. Each system artifact is linked to a specific aspect of COSYSMO to reflect the implications of system architectures on systems engineering.

Parametric Cost Modeling. The use of parametric models in engineering management serve as valuable tools for engineers and project managers to estimate person-month effort associated with specific activities. Developing these estimates requires a strong understanding of the factors that affect, in the case of this paper, systems engineering effort. Industry and academia have teamed up in the past to develop one of the most popular software development models in the world: the Constructive Cost Model II, better known as COCOMO II. Continuing in this tradition, the Constructive Systems Engineering Cost Model, or COSYSMO, represents the latest collaboration between CSE and its Corporate Affiliates. Leveraging off the strong relationships with industry, a working group was created to begin the development of the initial version of COSYSMO and identify possible sources of data to use for calibration of the model. The diverse experience of the working group members includes but is not limited to space systems hardware, information technology, radars, satellite ground stations, and military aircraft. This broad scope ensured that the model is robust enough to address multiple technical domains.

The typical involvement of affiliate companies is twofold. First, each company provides a group of systems engineering experts to rate the model drivers through the use of a wideband Delphi survey. This exercise allows for expert judgement to be captured and included in the model . An additional source of expertise has been the members of INCOSE who have provided extensive valuable feedback that has greatly improved the model. Second, the Affiliate companies provide historical project data for the COSYSMO calibration to validate the model parameters. This ensures that the Cost Estimating Relationships (CERs) in the model are appropriately weighed according to the data received from completed projects.

Industrial participation in the development of COSYSMO is key to the usefulness and relevance of the model. The initial industry calibration provided a validation of the model's robustness, establishment of initial relationships between parameters and outcomes, and determining the validity of drivers. In order to improve the accuracy of the model, each organization using COSYSMO will need to perform a local calibration. Through the industry calibration, the working group can establish baseline values for various scale factors for each driver. This might not be possible or feasible from a local calibration due to the size of the calibration data set and the narrow scope of a single organization's project database. The industry data can also identify elements or features of the model that need refinement to ensure that the model is generalizable across different domains in which systems engineering is applied. Obtaining data from multiple sources may also identify new drivers that need to be included in future revisions of the model.

An additional important reason for an industry-level calibration is the acceptance of the model for cost estimation by the Defense Contract Audit Agency (DCAA). Even though each organization needs to prove the local calibration matches the local organization's productivity and trends, the industry calibration shows DCAA the model meets the expectations and standards of the Systems Engineering industry. Ensuring that COSYSMO is compatible with these standards plays an important role in its widespread acceptance and generalizability.

COSYSMO and DoDAF

The COSYSMO model is made up of four size drivers and fourteen effort multipliers. The size drivers attempt to represent the functional size of a system to determine how large the system is in terms of systems engineering. The effort multipliers represent context parameters that capture the complexity of the project environment under which the systems engineering team is operating. The current set of drivers was refined through three rounds of Delphi surveys and validated through the use of historical project data (Valerdi 2005).

Size Drivers. The size drivers are quantitative parameters that can be derived from project documentation. Table 1 lists the typical sources that can provide information for each of the four size drivers in COSYSMO.

Table 1: Size Drivers and Corresponding Data Items

Driver Name	Data Item
Number of System Requirements	Counted from the system specification
Number of Major Interfaces	Counted from interface control document(s)
Number of Critical Algorithms	Counted from system spec or mode description docs
Number of Operational Scenarios	Counted from test cases or use cases

Early in the system life cycle, these sources may not be available to organizations due to the evolutionary nature of systems. In this case surrogate sources of data must be obtained or derived in order to capture leading indicators related to the four size drivers. Some of these sources may be previous acquisition programs or simulations of future programs.

Each size driver takes the form of both a continuous and categorical variable. As a continuous variable it can represent a theoretical continuum such as “requirements” or interfaces”, which can range from small systems to very large systems of systems; with most cases falling within an expected range. As a categorical variable it can be represented in terms of discrete categories such as “easy” or “difficult” that cannot be measured more precisely. The categorical scales are presented next and the counting rules for determining the values of the continuous variables are provided in the following sections.

Each size driver can be adjusted by three factors: complexity, volatility, and reuse. The total sum of requirements is adjusted upwards according to how complex each requirement is and how much the total requirements increase from the initial baseline. The sum of requirements is adjusted downwards when there are a significant number of reused requirements.

Cost Drivers. The cost drivers in the model represent the multiplicative part of the COSYSMO model. These drivers are also referred to as effort multipliers since they affect the entire systems engineering effort calculation in a multiplicative manner. Assigning ratings for these drivers is not as straight forward as the size drivers mentioned previously. The difference is that most of the cost drivers are qualitative in nature and require subjective assessment in order to be rated. Table 2 shows the data items or information that is needed in order to rate the cost drivers.

Table 2: Cost Drivers and Corresponding Data Items

Driver Name	Data Item
Requirements understanding	Subjective assessment of the system requirements
Architecture understanding	Subjective assessment of the system architecture
Level of service requirements	Subjective difficulty of satisfying the key performance parameters
Migration complexity	Influence of legacy system (if applicable)
Technology risk	Maturity, readiness, and obsolescence of technology
Documentation to match life cycle needs	Breadth and depth of required documentation
# and Diversity of installations/platforms	Sites, installations, operating environment, and diverse platforms
# of Recursive levels in the design	Number of levels of the Work Breakdown Structure
Stakeholder team cohesion	Subjective assessment of all stakeholders
Personnel/team capability	Subjective assessment of the team's intellectual capability
Personnel experience/continuity	Subjective assessment of staff consistency
Process capability	CMMI level or equivalent rating
Multisite coordination	Location of stakeholders and coordination barriers
Tool support	Subjective assessment of SE tools

In the process of collecting data to calibrate COSYSMO many lessons were learned (Valerdi et al 2004) that helped us improve our data collection strategy. One lesson learned was tied to the identification of entities and attributes that would be helpful in identifying COSYSMO size drivers. In this context, entities are defined as: "the object that is measured" and attributes are "a distinguishable property or characteristic of the entity" (Miller 2004). In the case of the COSYSMO size drivers, the *# of Major Interfaces* driver could be counted by identifying the number of Interface Control Documents (ICDs) as the entity and the number of interfaces documented as the attribute. In the case where a system is still in the early stages of development and no ICDs have been defined, an alternate method of counting interfaces is by using the DoDAF Operational View as the entity and the number of lines or arrows representing an interface as the attribute. Similar uses of DoDAF apply to the COSYSMO drivers and are identified in Table 1.

Driver Name	*Potential Architecture Products*
Number of System Requirements	The architecture development process and the requirements engineering process are interrelated and fairly loopy. The cost modeler can benefit from this iterative exercise.
Number of Major Interfaces	As mentioned earlier, the number of interfaces in the Interface Control Document helps provide a ballpark figure for this size driver. DoDAF products, notably, **SV-1** ("System Interface Description") and **SV-2** ("System Communication Description"), the **OV-3** ("Operational Information Exchange Matrix") can also help in identifying the total number of major interfaces.
Number of Critical Algorithms	The operational and the system view have specific architectural products which can help in estimating the number of critical algorithms. Notably, from the operational view, the **OV-6a, b and c** ("Operational Rules, State Transition and Event-Trace Description") and the **SV-10a, b and c** ("System Rules, State Transition and Event-Trace Description") from the system view can help in understanding and estimating the number of critical algorithms.
Number of Operational Scenarios	The **OV-1** "High-level Operational Concept Graphic" maps to use-cases when architecture products are developed using the Object-Oriented methodology. The total number of use-cases can be used in estimating this size driver.
Requirements Understanding	Similar to "Number of System Requirements"
Architecture Understanding	There are several architectural products which can help in quantifying this effort multiplier. Ideally, observing all the architectural products can help the cost modeler develop a better understanding and help in quantifying the "*subjective assessment*". In particular, the following products are suggested to be useful: **AV-1** ("Overview and Summary Information") and **AV-2** ("Integrated Dictionary") can help the cost modeler understand the basic motivations and objectives behind the architectural and also the system development effort. The **OV-1** ("High-Level Operational Concept Graphic") gives a high-level description of the operational concept of the system of interest. The **OV-5** ("Operational Activity Model") provides a description of the operational activities and the relationships amongst these activities, simultaneously, the **SV-4** ("System Functionality Description") provides a similar depiction however from the perspective of a system's operations. Both of these are amalgamated in the **SV-5** ("Operational Activity to Systems Function Traceability Matrix"). Together these architectural products have the potential to help the cost modeler develop an intuitive understanding not just into the cost modeling effort but also the architectural effort required for system design and development. **SV-6** ("System Data Exchange Matrix) and the SV-11 ("Physical Schema") along with the **OV-7** ("Logical Data Model") can also assist with this effort.

Documentation to Match Lifecycle needs	**AV-1** (“Overview and Summary Information”) and **AV-2** (“Integrated Dictionary”) lead to the development of documentation which matches lifecycle needs.
Migration Complexity	**SV-8** (“System Evolution Description”) describes the incremental steps required in migrating an existing system or suite of systems to a future implementation.
Technology Risk	There are two products with the potential to predict technology risk. Firstly, SV-3 (“System-Systems Matrix”) depicts the functions performed by various systems, secondly, SV-9 (“System Technology Forecast”) describes the emerging software/hardware technologies expected to be available in the near-term future with the potential to affect the existing system architecture.
Number and Diversity of Installations/Platforms	There are at least three architectural products which can help in gaining a deeper understanding of this effort multiplier. Firstly, **SV-1** (“System Interface Description”) describes the interconnections between and within the system items and nodes and systems. Secondly, **SV-2** (“System Communication Description”) describes how the systems and their items and nodes communicate with each other. Lastly, **SV-4** (“System Functionality Description”), describes the functions the systems perform and the flow of data through them. Together, these three products help understand the number and diversity of installations and platforms.
Number of recursive levels in the design	The Work Breakdown Structure, is closely related to the functional architecture in a systems engineering effort. Particularly, the Operational View products in DoDAF are quite useful in estimating this effort multiplier.
Stakeholder Team Cohesion	The **OV-4** ({Organizational Relationship Chart”) describes the relationships between the various participating organizations. While this does not necessarily explain the cohesion between stakeholder teams, it certainly helps in understanding the various organizational elements involved in the systems engineering effort.
Personnel/Team Capability	The DoDAF “Deskbook” discusses the role of humans in architectures. In particular, the **OV-5** (“Operational Activity Model”) can be transformed into “Human Activity Model” describing human functions, tasks and responsibilities. The **SV-4** (“System Functionality Description”) and the **SV-5** (“Human activities to operational activities traceability matrix”) can also be developed from the human point of view. While these products can’t explicitly express personnel capabilities, it can at least provide the cost modeler an
Process Maturity	**TV-1** (“Technical Standards Profile”) and **TV-2** (“Technical Standards Forecast”) describe the existing and emerging standards for the System View elements and hence can help obtain insights into the existing process maturity level. **SV-8** (“System Evolution Description”) can also be used to develop Capability Maturity roadmaps and can help the cost modeler to obtain insights into the

	process maturity. **SV-7** ("System Performance Parameter Matrix") can also prove useful in assessing the process maturity.
Multi-site Coordination	Two DoDAF products **OV-2** ("Operational Node Connectivity Description"), and **OV-3** ("Operational Information Exchange Matrix") can help in understanding the different operational nodes and thereby give insights into the coordination required in a multi-site effort. Along with these two products the **SV-3** ("System-Systems Matrix") also has the potential to help the cost modeler in estimating this effort multiplier.
Tool Support	The tools used in the architecture development effort, can provide a reasonable estimate for this effort multiplier.

Table 3: COSYSMO Size Drivers and Effort Multipliers Mapped to DoDAF Products

As can be inferred from the above table, all 26 products in DoDAF are useful in measuring or estimating size or cost drivers. Some architecture products such as the SV-2, 3, 4 or the OV-3 find use in estimating multiple cost or size drivers. In the section that follows, a case study from literature is chosen to depict how the architecture products can be mapped to the cost and size drivers in COSYSMO.

System Example: Oil Co's FastPass

To illustrate the benefits of integrating COSYSMO and DoDAF products the OilCo FastPass system is used as an example. The system, developed at George Mason's Systems Architectures Laboratory, provides a rich context to frame the discussion of interest (Wagenhals et al 2000). It should be noted these products were developed with the Command, Control, Computers, Communication, Intelligence, Reconnaissance and Surveillance (C4ISR) Architecture Framework.

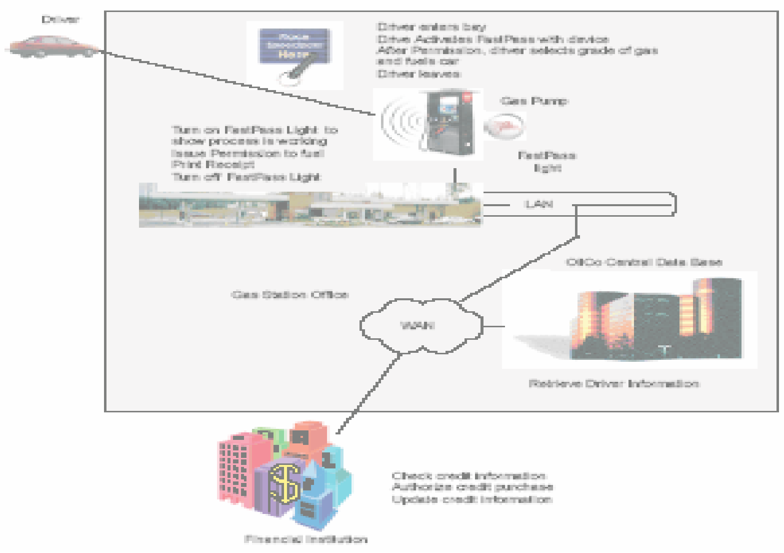

Driver
Driver enters bay
Driver Activates FastPass with device
After Permission, driver selects grade of gas and fuels car
Driver leaves
Gas Pump
Turn on FastPass Light to show process is working
Issue Permission to fuel
Print Receipt
Turn off FastPass Light
FastPass light
LAN
OilCo Central Data Base
Gas Station Office
WAN
Retrieve Driver Information
Check credit information
Authorize credit purchase
Update credit information
Financial Institution

Figure 1. TBD

(Wagenhals, Shin, Kim and Levis 2000)

To give a brief background of the FastPass system, Oil Co is a major oil company and FastPass is a new technology developed by Oil Co. Customer information is encoded in a small chip capable of receiving and transmitting radio signals. Customers sign up for it after providing standard information such as name, credit card, and current address. Figure 1 represents an instantiation of the system in function. When a customer enters the Oil Co gas station sensors at the pump sense the presence of a "FastPass" tag. Information from the tag is retrieved and sent to the credit company for authorization If the card is authorized, the pump is enabled else it stays disabled and the "FastPass" light goes off. Drivers are asked to select the grade of the gas and on selection the pumping begins. On completion the cost of the transaction is verified and sent to the credit company for accounting and updating. Data about the sale is entered in the gas station electronic ledger simultaneously.

In order to explain the symbiotic relationship between COSYSMO and DoDAF, this example focuses on one effort multiplier namely: "Multisite Coordination".

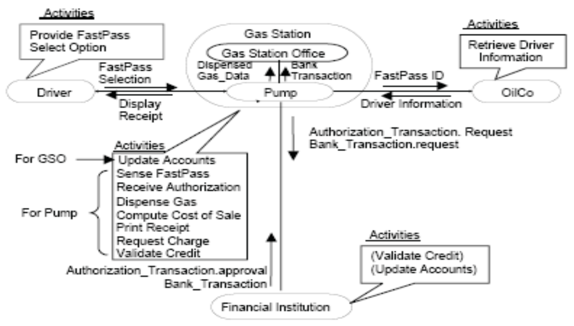
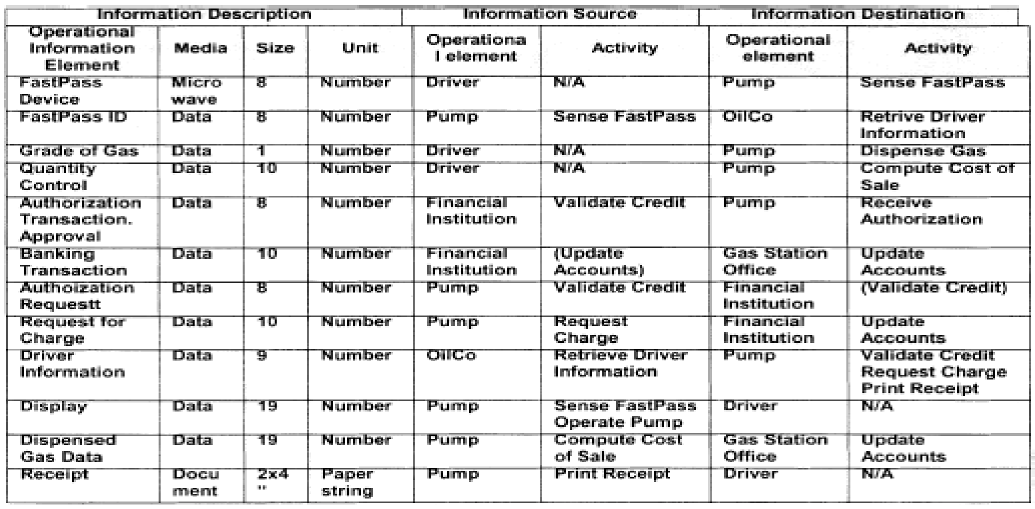
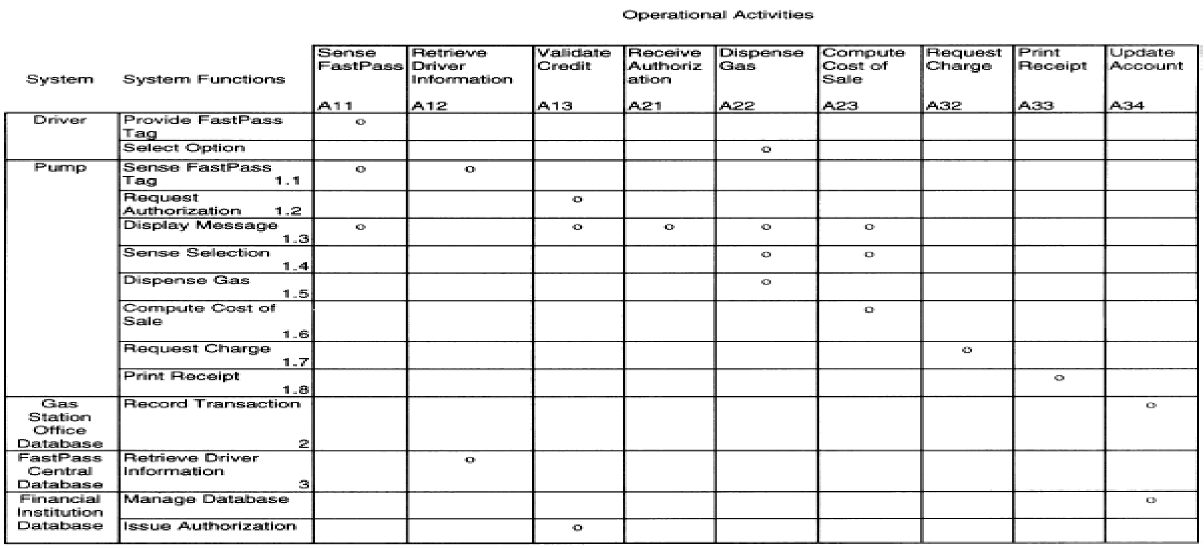


Figure 2. OV-2
(Wagenhals, Shin, Kim and Levis 2000)



Information Description				Information Source		Information Destination	
Operational Information Element	Media	Size	Unit	Operationa l element	Activity	Operational element	Activity
FastPass Device	Micro wave	8	Number	Driver	N/A	Pump	Sense FastPass
FastPass ID	Data	8	Number	Pump	Sense FastPass	OilCo	Retrive Driver Information
Grade of Gas	Data	1	Number	Driver	N/A	Pump	Dispense Gas
Quantity Control	Data	10	Number	Driver	N/A	Pump	Compute Cost of Sale
Authorization Transaction. Approval	Data	8	Number	Financial Institution	Validate Credit	Pump	Receive Authorization
Banking Transaction	Data	10	Number	Financial Institution	(Update Accounts)	Gas Station Office	Update Accounts
Authoization Requestt	Data	8	Number	Pump	Validate Credit	Financial Institution	(Validate Credit)
Request for Charge	Data	10	Number	Pump	Request Charge	Financial Institution	Update Accounts
Driver Information	Data	9	Number	OilCo	Retrieve Driver Information	Pump	Validate Credit Request Charge Print Receipt
Display	Data	19	Number	Pump	Sense FastPass Operate Pump	Driver	N/A
Dispensed Gas Data	Data	19	Number	Pump	Compute Cost of Sale	Gas Station Office	Update Accounts
Receipt	Docu ment	2x4 "	Paper string	Pump	Print Receipt	Driver	N/A

Figure 3. OV-3
(Wagenhals, Shin, Kim and Levis 2000)



		Operational Activities								
System	System Functions	Sense FastPass A11	Retrieve Driver Information A12	Validate Credit A13	Receive Authorization A21	Dispense Gas A22	Compute Cost of Sale A23	Request Charge A32	Print Receipt A33	Update Account A34
Driver	Provide FastPass Tag	o								
	Select Option					o				
Pump	Sense FastPass Tag 1.1	o	o							
	Request Authorization 1.2			o						
	Display Message 1.3	o		o	o	o	o			
	Sense Selection 1.4					o	o			
	Dispense Gas 1.5					o				
	Compute Cost of Sale 1.6						o			
	Request Charge 1.7							o		
	Print Receipt 1.8								o	
Gas Station Office Database	Record Transaction 2									o
FastPass Central Database	Retrieve Driver Information 3		o							
Financial Institution Database	Manage Database									o
	Issue Authorization			o						

Figure 4. SV-4

(Wagenhals, Shin, Kim and Levis 2000)

As mentioned earlier "Multisite Coordination" involves understanding the location and coordination of stakeholders. As can be observed from Figures 2, 3 and 4, there are clearly three sites which need to coordinate with each other – the gas station, the banking or financial institution and Oil Co and lastly, the human interest, the individual seeking to fill up gas in his or her vehicle. The modeler thereby can easily incorporate these details into cost model, thereby helping to quantify subjective information in the COSYSMO cost driver. While this may seem to be a trivial example, it quite clearly elucidates the use of architectures in the cost modeling effort.

Conclusions

This paper has presented ways that the DoDAF can help estimate systems engineering cost through the use of the Systems Engineering Cost Model (COSYSMO). The use of the Operational View provided rich information about the interfaces and operational scenarios; the System View painted a clearer picture from the system algorithms standpoint; and overall the DoDAF products provided useful information at the enterprise level to help understand the systems engineering cost implications of the architecture. Capturing this information can lead to more accurate cost estimates since all of these parameters are highly correlated to systems engineering effort, and ultimately cost. The goal is for COSYSMO to better represent the system architecture using techniques like these so that systems engineering can be estimated with a higher degree of accuracy.

One interesting finding was that we were unable to map three of the subjective cost drivers in COSYSMO, highlighting limitations in the realm of "soft systems engineering" not covered by DoDAF. Namely, requirements understanding, number of recursive levels in the design, and tool support. Regardless of the limitations, we hope that using DoDAF for cost estimation should gives organizations an added impetus for the practicality of DoDAF beyond just the modeling aspects.

References

Bienvenu, M. P., Shin, I., Levis, A. H., C4ISR Architectures: III. An Object-Oriented Approach for Architecture Design, *Systems Engineering*, Vol. 3, No. 4, 2000.

Valerdi, R., Rieff, J., Roedler, G., Wheaton, M., "Lessons Learned for Collecting Systems Engineering Data," 2nd Annual Conference on Systems Engineering Research, Los Angeles, CA, April, 2004.

Valerdi, R., "The Constructive Systems Engineering Cost Model," unpublished PhD dissertation, University of Southern California, August 2005.

Wagenhals, L. W., Shin, I., Kim, D., Levis, A. H., C4ISR Architectures: II. A Structured Analysis Approach for Architecture Design, *Systems Engineering*, Vol. 3, No. 4, 2000.

DoDAF, Department of Defense Architecture Framework, Deskbook.

Biography

Ricardo Valerdi is a Research Associate at the Lean Aerospace Initiative at MIT and a Visiting Associate at the Center for Software Engineering at USC. He earned his BS in Electrical Engineering from the University of San Diego, MS and PhD Systems Architecting & Engineering from USC. Formerly he was a Member of the Technical Staff at the Aerospace Corporation in the Economic & Market Analysis Center and a Systems Engineer at Motorola and at General Instrument Corporation.

Indrajeet Dixit is a doctoral student in the Daniel J. Epstein Department of Industrial and Systems Engineering at USC. Prior to becoming a "Trojan" he acquired an MS in Systems Engineering from George Mason University, Virginia in 2003 and a BS in Electronics Engineering from Mumbai (Bombay) University, India in 2000.