SAND 2008-7041C

Agile Development of Tractable Analyses and Simulations of Complex Systems

Robert J. Cloutier
School of Systems and Enterprises
Stevens Institute of Technology
Hoboken, NJ 07030

Mark J. De Spain, John M. Linebarger, Floyd W. Spencer
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185

Abstract. Agile development differs from other engineering processes in the manner in which each task is performed, and the ability to respond to changes in scope or requirements. The agile model permits continual feedback after a complete pass through each of the disciplines. Agile development enables the system to be built in a series of cycles from a set of rudimentary capabilities to the full system capability. When agile development is applied to the analysis of complex systems and "wicked problems," the inherent conflicts and inconsistencies of those systems can be resolved. The result of applying "agile" to analysis of a wicked problem is the Design for Tractable Analysis (DTA) framework. DTA analyzes the system (or enterprise) of interest as a whole, in conjunction with decomposing the system into constituent elements for domain specific analyses that are informed by the whole. The use of DTA is demonstrated through a case study of a complex security system.

Introduction

Systems analysis approaches to persistently challenging problems, which have a variety of stakeholders and scenarios, are traditionally solved using linear or canonical methods, and can often take many months to set-up and complete. Often, there is no project plan. Using traditional systems engineering approaches, the team will decomposes the overall system into subsystems where most of the analysis effort is then applied. Effects due to interdependencies between the subsystems are often not analyzed to the same depth as the subsystems primarily because there is no good closed form analytic method for understanding the contributions of subsystem interactions.

However, as the demand has grown for more capable and autonomous systems to address increasingly complex geopolitical, environmental, security, and combat environments, the problem space has transitioned from linear problems into the "wicked problem" domain, where the solution actually transforms the problem (Hodge and Weinberger, 2005). To make it even more challenging, the demand for these complex systems occurs in environments of evolving, or even rapid changing requirements and timelines. Therefore, the need for an agile approach to systems engineering is becoming more and more critical. Within the software community, there are a number of different agile methodologies, including:

- Scrum
- Extreme programming (XP)
- Agile Modeling
- Crystal
- Lean Development
- Rational Unified Process (RUP)
- Agile Unified Process (AUP)

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

- Open Unified Process (OpenUP).

The important characteristics of all of these are that they promote iterative development in teams and adaptive collaboration throughout the life-cycle. Agile processes desire to deliver small, incremental, and demonstrative capability, working to the final product. Usually the project is broken into phases that promote understanding of the problem while advancing the product from phase to phase. High-risk items are identified and worked early in the project. Teams are cross-functional, and tend to work together through the life of the project.

The Rational Unified Process (Rational, 2001), an agile process which can be used to address wicked problems, was commercialized by the Rational Software Company before being purchased by IBM. Figure 1, sometimes called a "sand" chart, is from RUP (Cantor, 2003), but very similar versions are found in any of the so-called "unified process" variants. In this approach, a project is broken down into phases, and each phase may have any number of iterations. One pass through Figure 1 can be called an increment or spiral. Each increment performs the normal development life-cycle tasks (called "Disciplines" in Figure 1), from mission modeling and requirements development through test. A short review and demonstration of capability is performed at the end of each iteration to demonstrate progress, and to assess the overall project health and status, allowing for adjustments as necessary.

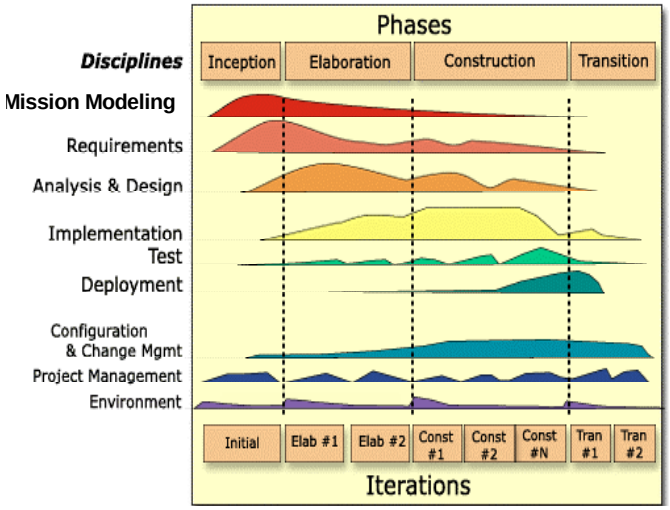

Figure 1. The Unified Process

Agile Systems Engineering. What differentiates an agile approach from other engineering processes is the manner in which each of those tasks is performed, and the ability to respond to changes in scope or requirements. The agile model permits continual feedback after a complete pass through each of the disciplines. Closer inspection of Figure 1 shows how the emphasis on particular disciplines varies over time. For example, early iterations spend more time on requirements and later iterations spend more time on implementation and test. The four phases - Inception, Elaboration, Construction, and Transition - structure the lifecycle of the development effort. If it is being used on larger projects, then the project can be broken into multiple increments (or spirals), each of which comprises all four phases (Figure 2).

Agile development enables the system to be built in a series of cycles from a set of rudimentary capabilities to the full system capability. Simulations and prototypes are used to define the approach or a proof of concept. Risk definition and management are keys to achieving success with this approach.

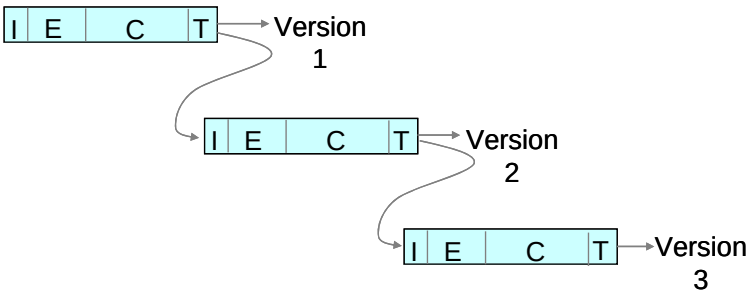

Figure 2. A Single Project with Multiple Increments

At the end of each Phase, there is a significant review. Figure 3 shows the major objectives, milestones, and artifacts for each phase. An assessment of an increment's success in meeting its objectives and the sufficiency of its artifacts is performed three times during the increment. The names of these reviews have changed over time, and while Figure 3 calls them Lifecycle Objective Review, Lifecycle Architecture Review, and Engineering Assessment, they can be named as desired. What is more important is the intent of each review and the nature of the artifacts produced.

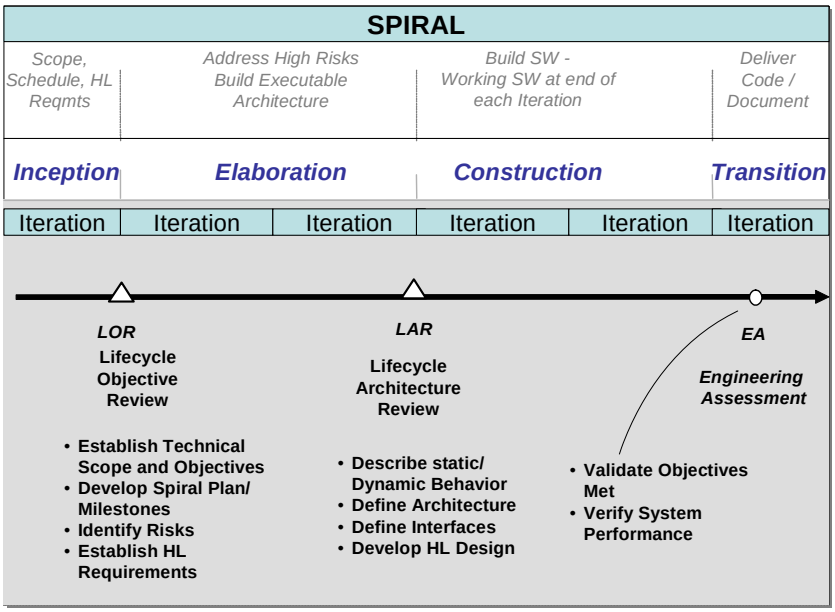

Figure 3. Formal Reviews and Objectives for an Increment

Though the Inception phase will be discussed later, it is important to note that the complete increment and the corresponding iterations are planned in broad terms during inception. The Project Plan defines the major milestones and the key artifacts in the spiral, along with their objectives, within each phase. The schedule and budget for the spiral and the required resources are also planned at a rough order of magnitude.

Iterations. Each of the phases is executed through a succession of iterations. The iterations are essentially small waterfall developments through the sand chart disciplines. A portion of the increment functionality will be analyzed, designed or built in each iteration. Iteration planning takes place at the beginning of each iteration, and is done concurrently with the assessment of the previous iteration to facilitate plan coordination. Iteration Plans implement the required Project Plan capability assigned for that period. Iteration Plans identify the system capabilities to be developed during the iteration, establish the mitigation strategy to be used to manage risks, assign risk mitigation tasks to current and future phase iterations, and address process improvement needs based on lessons learned during previous iterations.

Design for Tractable Analysis (DTA): An Analysis Framework for Complex Systems. DTA (Linebarger *et al.*, 2009) relies on the adage "form follows function" by analyzing a system as a whole based on unifying behavioral domains, rather

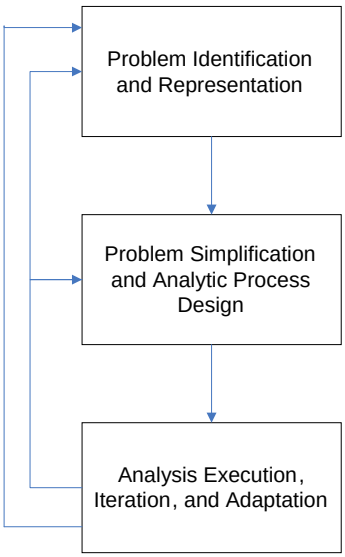

Figure 4. DTA Process Flow

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

than decomposing the system into constituent subsystems for independent analysis, commonly referred to as reductionism; Barton and Haslett (2007) present a good discussion of the tension between reductionism and reverse reductionism in science, which is characterized as a dialectic between analysis and synthesis. DTA is particularly good at untangling the interdependencies that characterize complex systems, and uses abstractions and patterns to exploit the way the analyst's mind works. The method frames the problem and the model in the context of the questions being asked by the stakeholder and the analyst. The nature of the questions the analyst asks about the system will affect the structure and execution of the model.

DTA is an iterative three-stage process (Figure 4). The stages are: 1) problem identification and representation, 2) problem simplification and analytic process design, and 3) analysis execution, iteration and adaptation.

Overview of the DTA Framework

Problem Identification and Representation. This paper will utilize a case study performed at Sandia National Laboratories over the course of 2007-2008 to demonstrate the DTA methodology. A use case diagram (Figure 5) represents a high-valued asset residing within a fixed-site facility. The problem requires that many types of operations be performed that are categorized either as mission critical operations or as facility-related operations. Additionally, there is a need to protect the asset from adversaries. Note that the use cases are directly derived from the stated purpose, which is to perform operations on a high-valued asset. The introduction of an adversary into the operations of the site prevents a deterministic solution to the security questions.

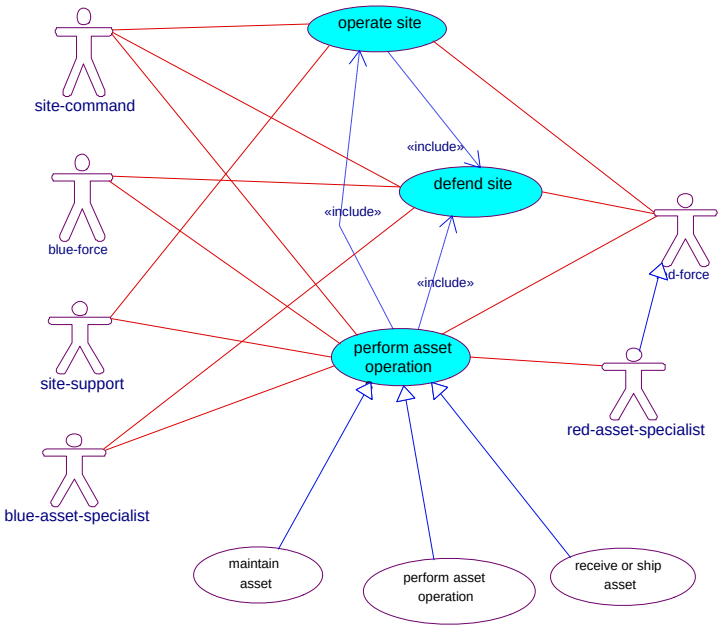

Figure 5. Top Level Use Case Diagram of Case Study

Figure 5 captures two different yet related views of the enterprise of interest on a single top-level Unified Modeling Language (UML) or Systems Modeling Language (SysML; Balmelli, 2007) use case diagram. One view is the operational design view represented by the "bubble" and "actor" elements and the other is the operational dependency view represented by the dashed dependency lines that are stereotyped with "include" in the diagram. The design view is necessary to specify the behaviors and structures in the enterprise and their relationships.

After the top-level use case diagram is created, it is hierarchically decomposed into additional lower-level use case diagrams. Figure 6 depicts a first-level decomposition of one of the tasks, Operate Site, from the top-level use case diagram.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

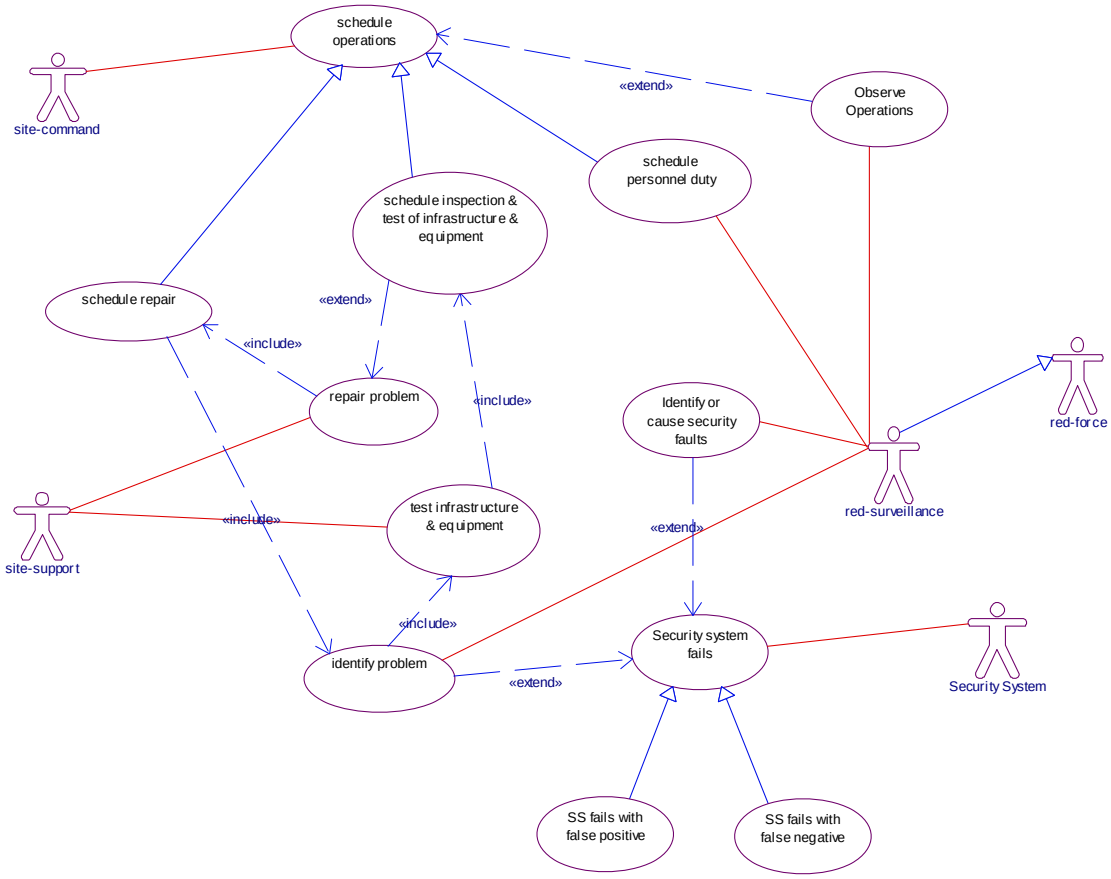

Figure 6. Second Level Use Case Diagram of Operate Site Activity

Applying an Agile Process to DTA

This section will look at each phase of an agile process, discussing the primary goals, using DTA to demonstrate the application. The paper will also perform double-duty in that we will discuss the DTA process itself, as well as its findings in the included case study.

The Inception Phase. The overriding goal of the inception phase is to achieve concurrence among all stakeholders on the objectives for the increment. The inception phase is focused on developing the objectives for the project by addressing the mission, requirements, and risks.

The primary objectives of the inception phase include:

- Setting the project scope and boundary conditions, including acceptance criteria and what is intended to be in the project and what is not.
- Discriminating the critical objectives of the project.
- Estimating potential risks (the sources of unpredictability)
- Preparing the supporting environment for the project.

Inception – Project Planning and Scoping. During inception for the DTA project, the project scope and project plan was developed and agreed upon by the stakeholders. Stakeholders included executive sponsors, participating organizations, and research partners. At this point of the project, the details of the project plan were not expected to be completely defined, just a high level definition of the project, the participants, the expectations, and a rough timetable, along with a definition of the goals of the project and budget. The beginnings of the statement of work were established, defining key project objectives and measures of effectiveness. While a risk management plan should be put in place, it must take on a scope that is commensurate with the size of the project. If the project requires infrastructure to be put in place, this should also begin during inception. Another consideration is whether

subcontractors will be needed. If subcontracts are necessary, the initial agreements for startup funding can be put in place, with the understanding that the complete statement of work (SOW) will be provided at the end of the elaboration phase. In this project, the UML/SysML diagrams were used to understand the scope of the project.

The inception phase ends with a review of the project objectives (Lifecycle Objectives Review). The goal is to ensure the project scope is understood and agreed to by all stakeholders, that risks have been indentified and mitigation plans are in place, and that all participants and stakeholders agree on the high level requirements of the project. If these are not satisfied, adjustments must be made at the beginning of the elaboration phase to ensure the project goals will be satisfied.

The Elaboration Phase. The goal of the elaboration phase is to baseline the architecture of the system in order to establish a stable basis for the majority of the design and implementation effort in the construction phase. The architecture evolves through the consideration of the most significant requirements (those that have a great impact on the architecture of the system) and an assessment of the associated risks. The architecture is evaluated through one or more architectural demonstrations during the elaboration phase.

The primary objectives of the elaboration phase include:

- Ensuring the architecture, requirements and plans are defined sufficiently to proceed, and the risks sufficiently identified and/or mitigated to be able to predict with some confidence, the cost and schedule for the completion of the development;
- Addressing the identified architecturally significant risks of the project;
- Establishing a baseline architecture by addressing the architecturally significant scenarios, which typically expose the top technical risks of the project; and
- Producing necessary demonstrators or evolutionary prototypes and models. These may be throw-away prototypes which are created to mitigate specific risks.

Project Name		<NOTE>	1. Enter the name of a project in cell '2A'.
Fixed Site Security Analysis			2. Enter all task names in column 'A'.
Task Name		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
O_observe operations - assess attack potential	1					2			2																								
O_adversary identifies/causes security fault	2	2																															
O_security system fails	3		1																														1
O_schedule inspection and test	4					1			1	1				1																			1
O_test and inspect	5			1	1																												
O_Identify problem	6					2																											
O_Schedule repair	7						1																										
O_repair problem	8							1		1																							
O_schedule personnel duty	9				2			2																									
P_formulate site defense plan	10											1		1																			
P_perform cost analysis of plan	11										2																						
P_execute site defense plan	12										1	1																					
P_evaluate effectiveness of defenses	13										1		1		1																		
P_run force-on-force exercises	14													2																			
M_monitor for attack: alarm triggered	15																1	2															
M_monitor for attack: alarm not triggered	16		1																1					2		2						2	
D_execute attack	17																								2	1							
D_assess situation	18															2				2	2						2			2			
D_deploy forces (blue)	19																	2	2														
D_neutralize adversary	20																			2													
A_protect asset	21																		2	2	2								2				
A_coordinate operation with support and blue-force	22																															1	
A_increase pro-force when bunker opened	23																						1										
A_observe operations - attack ready	24		1	1																													
A_access asset	25																							1	2								
A_monitor security of asset	26																							2		2	2			2	2		
A_perform operation on asset	27																									1							
A_threaten asset	28																	2															
A_secure asset if under threat	29																									1			2				
A_store asset in protected bunker	30																									1				2			
A_schedule asset operation	31								1					1																			
O_determine system availability	32				2				1		2																						

Figure 7. Fixed-Site Design Structure Matrix (DSM) Input Matrix

Elaboration - Problem Simplification and Analytic Design. During the elaboration phase for DTA, the analyst designs an analytic process that will tractably address the key analytic

questions. UML/SysML has already informed the analyst of the processes and relationships that exist in the modeled enterprise. Using a Design Structure Matrix (DSM), the analyst then simplified the problem and analytic process. We call this step "designing the analysis" because not only does it identify the analysis tools needed, but it also indicates the order in which they should be applied.

Figure 7 depicts the input to the DSM matrix (Cho, 2007) for the fixed-site pilot scenario. Each of the entries on the left is a leaf-node task from the decomposition of the use case diagram in Figure 6. There is no specific order required for entering the values into the matrix. The distinction between '1' and '2' in the cell entries is that '1' represents a sequential ordering where the initiating task must be completed prior to the start of the trailing task, and '2' indicates a data dependency where the trailing task can start any time after the start of the initiating task. What is important, however, is to make sure the relationships are completely identified. As is shown in Figure 3, multiple iterations might be necessary during the elaboration phase. This may also be true for the models created during the elaboration phase. In practice, the authors found that several iterations were required to ensure that the information is correct and complete in the DSM.

In Figure 8 the input DSM of Figure 7 has been transformed by reordering the tasks to minimize the number of feedback relationships. Those that do remain are grouped into self-contained iterative operational blocks. The tasks included in each block represent a set of related activities that must be iterated to derive a useful result for downstream tasks. The tasks that are not contained within an iterative block are essentially sequential, although some may be performed in parallel if they fall within the same level of the transformed matrix. For example, tasks 21 and 22 appear to be out of order, but they are actually executed in parallel; in other words, based on the scenario for the fixed site, the adversary can launch the attack when the operation on the asset has begun.

Task Name	Level		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
P_formulate site defense plan	1	1		1		1																													1
P_perform cost analysis of plan	1	2	2																																2
P_execute site defense plan	1	3	1	1																															3
P_evaluate effectiveness of defenses	1	4	1		1		1																												4
P_run force-on-force exercises	1	5				2																													5
O_observe operations - assess attack potential	2	6									2				2																				6
O_adversary identifies/causes security fault	2	7						2																											7
O_security system fails	2	8							1								1																		8
O_schedule inspection and test	2	9				1						1			1	1	1																		9
O_test and inspect	2	10								1	1																								10
O_Identify problem	2	11										2																							11
O_Schedule repair	2	12											1																						12
O_repair problem	2	13												1		1																			13
O_schedule personnel duty	2	14									2				2																				14
O_determine system availability	2	15	2								2					1																			15
A_observe operations - attack ready	3	16							1	1																									16
A_schedule asset operation	3	17				1									1																				17
A_coordinate operation with support and blue-force	4	18																	1																18
A_increase pro-force when bunker opened	5	19																		1															19
A_access asset	6	20																2			1														20
D_execute attack	7	21																2				1													21
A_perform operation on asset	7	22																				1													22
A_threaten asset	8	23																					2												23
A_secure asset if under threat	9	24																				1			2										24
A_store asset in protected bunker	10	25																				1				2									25
A_monitor security of asset	11	26																			2	2		2		2	2								26
M_monitor for attack: alarm triggered	12	27																					2						1						27
M_monitor for attack: alarm not triggered	12	28							1										2		2	2													28
D_assess situation	12	29																								2		2	2				2	2	29
D_deploy forces (blue)	12	30																					2												30
D_neutralize adversary	12	31																															2		31
A_protect asset	12	32																							2						2	2	2		32
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	

Block1: Site Preparation

Block2: Logistics and Maintenance

Block3: Battle

Figure 8. Fixed-Site DSM Output Matrix

The analysis questions are dependent on the relationships between the use cases since that governs the order in which the questions are answered. Often the dependency-driven order may not be the order that is most important to the primary stakeholder.

For the case study, ordering between the tasks, as shown in Figure 5, is indicated by the "include" stereotype on the association lines that link the tasks. In other words, since "perform asset operations" includes, or is dependent on the completion of "operate site" and "defend

site," the latter two use cases take precedence temporally. Finally, since "perform site operations" depends on "defend site," the latter takes precedence. This order of precedence means that "defend site" is a necessary condition for the other two, which otherwise would be in jeopardy.

Placed in the order of dependence, and taking lifecycle costs from the DSM into account, the three use cases of "protect site and asset," "maintain site," and "perform mission critical operations," lead naturally to the following three key analysis questions:

1) What is the optimal balance of cost, technology and performance that maximizes the security of the site and asset while minimizing the deployment costs?
2) What are the optimal maintenance and personnel schedules that maximize site security readiness and minimize sustainment costs?
3) What is the maximum number of mission critical operations that can be executed with acceptable risk for a given budget?

Note that answering the third question completes the first iteration of the analysis, which determines the viability of the site as an enterprise.

Each succeeding question is dependent on the previous question, and interdependencies make it impractical to attempt to answer all questions concurrently. The DTA is designed to restructure the tasks of the analysis model and shift the focus of the analysis as answers are sought for each question. The consequence is that the analysis at the enterprise level proceeds sequentially with an iterative overlay when subsequent questions are not readily answered (see Figure 1). That is, several iterations may be required to balance the conflicting objectives of each question relative to the objectives of the other questions. To a first order this means that the qualitative words used in the above questions, such as "maximizes," "optimal," and "acceptable," do not reflect absolute valuations; instead, each must be balanced across all the objectives for the system.

Finally, the main interest of the primary stakeholder will generally revolve around how much value he derives from his dollar; that is, his focus is going to be on the last question regarding the number of operations that can be performed, given the associated risk and the overall cost of operations. The primary stakeholder may not be interested in the particulars of the first two questions; however, the concerns of the primary stakeholder cannot be adequately addressed without resolving the issues identified in the first two questions (*e.g.*, see Figure 5). In other words, the three analysis questions are sequentially dependent on each other.

Also defined during the elaboration phase was the project architecture (Figure 9) which consisted of a scenario for a notional force-on-force battle simulation at a notional fixed-site facility that could be an armed forces base, a nuclear power plant, an embassy compound or other secured facility. The analysis used a software application developed at Sandia called Dante (Design Analysis of Neutralization Technologies Evaluator), which supports the analysis of physical security systems. Sophisticated stochastic models are employed to represent the uncertainties associated with battle, which require numerous iterations to converge on a result. Dante simulates force-on-force engagements, and uses batch mode processing to analyze the large amount of data generated by a given scenario. Not only can the data be analyzed to compute the probability of neutralization, it is also used to gain additional insight into the effectiveness of the physical security system and potential options for improvement.

Also defined was the model integration and execution architecture. The project team used a combination of commercially available tools (Excel, iSight-FD) and internally developed tools (Dante). The final agreed-upon goal of the simulation was to discover the topology in both overall cost and battle effectiveness with respect to the number and placement of sensor grids, which monitor the fixed-site facility and its assets. A design of experiments module drives a loop which consists of the Dante application and a cost model currently implemented in Microsoft Excel (Figure 10).

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

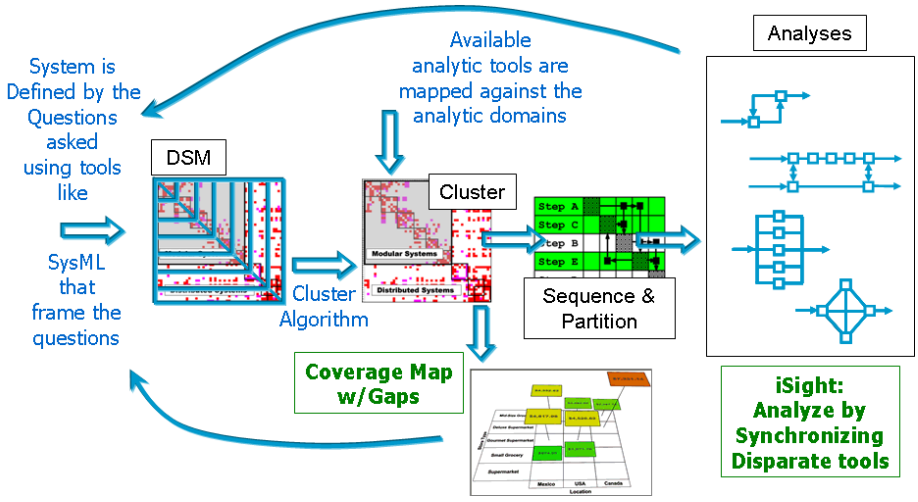

Figure 9. Design for Tractable Analysis (DTA) Architecture

The elaboration phase ends with a review of the project architecture. The goal is to ensure the established DTA architecture is appropriate to meet the goals of the project. If it not, adjustments are needed at the beginning of the construction phase to satisfy the project goals.

The Construction Phase. The goal of the construction phase is to clarify the remaining requirements and complete the bulk of the project based upon the baseline architecture. For the DTA approach, this included the construction of the analysis questions, and the tools to support the solving of those questions. Like elaboration, the construction phase will be broken into small iterations to ensure the ability to analyze the progress and make corrections as necessary to meet the goals of the project plan.

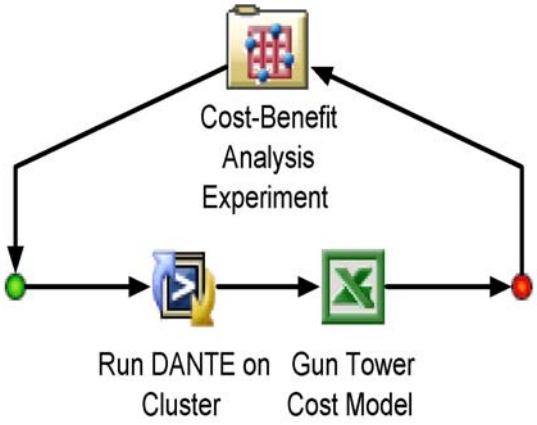

Figure 10. iSight Simulation Execution Workflow

The primary objectives of the construction phase include:

- Achieving adequate quality as rapidly as practical;
- Achieving useful versions as rapidly as practical;
- Completing the analysis, design, development, and testing of the required functionality as defined in the Project Plan;
- Iteratively developing a complete analysis that is ready to deliver. This implies describing the remaining use cases and other requirements, fleshing out the analysis design, completing the implementation, and testing the solution; and
- Validate the new system against user expectations.

As discussed earlier, during the elaboration phase the questions an analyst asks will affect the structure of the resulting analytic model. In the context of the case study of securing a fixed-site asset, we now show the process by which a good set of analysis questions are derived. In essence, the three tasks or use cases in the top-level use case diagram of Figure 5 formed the basis for three questions that will determine the analysis flow. In addition to the use cases, a DSM for the enterprise might also highlight relationships and constraints that affect the determination of analysis questions.

Construction – Constructing the Analysis Data Flow. The two major outputs of this analysis were a preliminary configuration for the sensor grids and a closed form equation that defines the effectiveness of grid configurations for the particular battle scenario. Needed are the derivations of similar equations of grid configuration effectiveness for other battle scenarios in order to identify a common sensor grid configuration for various scenarios. At that point the analysis can proceed to address the logistical/maintenance issues identified in the second question and finally proceed to answer the question that is uppermost in the mind of the stakeholder: "how many asset operations can be performed at what cost and risk?"

Dante was used to answer the first analysis question regarding optimization of site defense. The assumption is made that there are no issues related to availability of equipment and personnel, and that all required mission critical operations can be fulfilled as scheduled, which obviates the need for Logistics and Maintenance analysis. This assumption was an approximation of reality, which the analysis can iteratively improve with later simulations, but it does make the initial analytic "spiral" tractable.

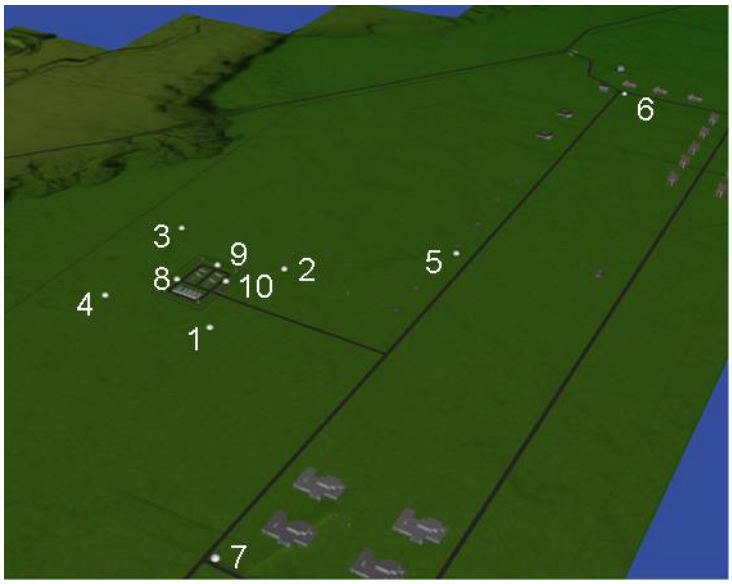

Figure 11. Positions for possible sensor grid

The decision that drives the case study is where to place grids of sensors in a sensor network, in order to cost-effectively protect the fixed site. The sensor grid contains a variety of sensor types, with a minimal amount of sensor fusion capability at a single location.

The Dante model parameters (independent variables) are the number and placement locations of the sensor grids (Figure 11). The primary response variable is the percentage of engagements won by the protective force (pro-force or the blue team; the attack force is called the red team). The cost model for this analysis is rather basic, since the deployment cost is essentially linear with each added sensor grid.

The basic attack scenario involved three red force snipers at the base of the hills near locations three and four as shown in Figure 11 (which is a screen shot from Dante), as well as a suicide convoy of four vehicles that enters the facility at the outer perimeter gate near location six and moves down to the road past location five. The convoy then moves into the inner perimeter at the gate in the fence near location ten. The objective is to gain entrance to an open bunker near location eight and take possession of an asset in the target bunker. There are pro-force personnel, vehicles and sensor grids at various locations throughout the facility that attempt to neutralize the attacking force, and to close the open bunker before the red force can penetrate the defenses.

In order to proceed to the next analysis question, the optimum security configuration should be determined, based on the deployment cost and the level of security provided by the sensor grids. If the cost is too high or the security inadequate, both of which are judgment calls dependent on stakeholder objectives, the analysis may have to start over and consider other changes to the system. These changes could involve adding new sensors, reconfiguring the facility, changing procedures, or making other alterations. The second analysis question is concerned with the availability of the various security subsystems. This issue involves a number of very thorny issues having to do with reliability, maintenance logistics, and even the potential for subversion. With this question, the analysis begins to move into the real world

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

where the situation is not always in an optimal state. This second question begins to address enterprise level concerns, where security may be affected by day-to-day operations. The issue is how to structure the operations of the site to ensure the maximum availability of the critical security subsystems. Again, costs play a role, because unlimited sustainment resources are not available to ensure that every piece of equipment and all personnel are always at their peak performance.

The analysis is now entering a phase where closure could be achieved; at the very least, additional data is now available to enhance the approximations used in the first question and allow refinement of the results in the next spiral. Proceeding with the final (third) question also addresses the purpose for implementing the fixed site, which is the performance of mission critical operations. The analysis of this question could be straightforward, depending on the conditions that drive the scheduling of mission critical operations. For example, if scheduling is flexible then the optimum security could be realized if all such operations were performed immediately after maintenance is completed on the most critical elements of the security system. (This conclusion assumes that the asset or site is most vulnerable to an attack during a mission critical operation due to the exposure created by an authorized access of the asset.) However, if the mission critical schedule is governed by external programmatic drivers, then the security risk may vary depending on the maintenance cycle. A judgment of whether the added risk is acceptable would have to be made by the stakeholder. If the risk is deemed to be too great then the analysis would have to revert back to either the prior logistics analysis, or even all the way back to the initial security configuration study.

The construction phase concludes with review of the implemented simulation infrastructure, a demonstration of the complete process flow through all of the tools, and sample data to ensure the system will produce output consistent with expectations.

The Transition Phase. The focus of the Transition Phase is to ensure that the product is ready. The Transition Phase can span several iterations, and includes testing the software in preparation for release, and making minor adjustments based on user feedback. At this point in the project lifecycle, user feedback should focus mainly on fine tuning the project, configurations, and installation and usability issues. All major structural issues should have been worked out much earlier in the project, and there should be no surprises at this point. By the end of the Transition Phase, all project objectives should have been met.

Activities performed during the Transition Phase depend on the goal. The Transition Phase is entered when products are mature enough to be delivered to the target baseline or customer. This typically requires that some usable subset of the product has been completed with acceptable quality level and user documentation so that transitioning to the user provides positive results for all parties.

The primary objectives of the Transition Phase may include:

- Testing to validate the new system against user expectations
- Testing and parallel operation relative to the previous project functionality
- Training of users and maintainers if necessary
- Tuning activities such as bug fixing, enhancement for performance and usability
- Assessment of the deployment baselines against the project objectives and the acceptance criteria
- Achieving stakeholder concurrence that the products are complete
- Achieving stakeholder concurrence that the products are consistent with the evaluation criteria

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

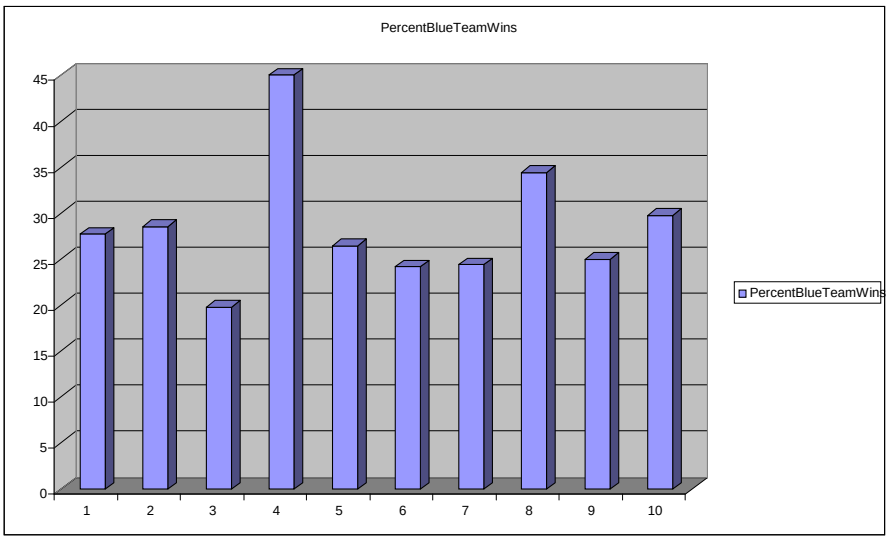

Figure 12. % blue wins for one sensor grid at location

Transition - Experiment Results. Since the purpose of the analysis of the pilot scenario was to evaluate the effectiveness of the sensor grids, the analysis began by determining the baseline performance when there were no grids present. In 480 runs of the baseline configuration (no sensor grids) runs in DANTE, the blue team stopped the red team 98 times for a "Percent Blue Team Wins" of 20.4%. The results when a single grid is added at any of the ten locations are shown in Figure 12.

In addition to performing the single grid analysis a larger number of sensor grids distributed throughout the ten locations of Figure 11 were analyzed. While all possible two grid configurations (45 in total) can be analyzed with a full factorial set of simulations, full factorials of larger grid configurations cannot be analyzed in a reasonable amount of time. Consequently, a balanced fractional factorial design matrix for the simulations was constructed to enable completion of the analysis in a reasonable time period that would also provide sufficient information to be able to estimate the performance of configurations not actually simulated.

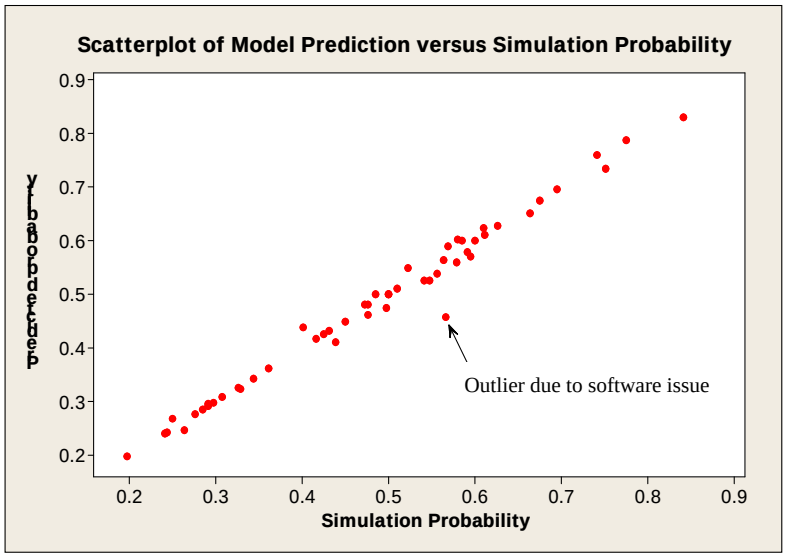

Figure 13. Predicted probability versus simulation

The results of the "balanced experiment" were analyzed statistically. A predictive model (Figure 13) was generated that estimates the highest probability of blue team wins for each number of sensor grids, and suggests which configuration of sensor grid locations yields the highest probability of wins for that number of sensor grids. The predictive model was validated by generating analytic results for several sensor grid location configurations not previously run.

Table 1 gives the outcomes of applying the predictive equation for combinations of four and three grid sets. The results show that the configuration of sensor grids at locations 3, 4, 9, and 10 would be expected to give the best results when four sensor grids are present. However, the potential of losing sensor grids 4 and/or 10 through mechanical failure, sabotage or other adversary action would lead to the conclusion that a network of sensor grids at locations 1, 4, 9, and 10 is preferable, because of the greater robustness if a grid is lost.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

	Probability of success			
SensorGrids included	All 4 present	Loss of SensorGrid04	Loss of SensorGrid10	Loss of SensorGrid04 and SensorGrid10
3,4,9,10	0.829	0.548	0.548	0.274
1,4,9,10	0.813	0.585	0.620	0.347
3,4,7,10	0.807	0.398	0.615	0.239
1,3,4,10	0.795	0.500	0.584	0.310
4,7,9,10	0.793	0.431	0.603	0.272

Table 1. Probability of Blue Team success with loss of one or more sensor grids

Cost-Benefit Analysis. Cost data was also calculated using a very simple notional cost model that took several factors into account: the cost of the sensor grid; the lifetime of the sensor grid; and the annual cost of maintenance, upgrade and test for each grid. The prediction equation was also used to calculate a return on investment. Figure 14 shows the results of the analysis and shows the relationship of the costs associated with having one through ten sensor grids available versus the maximum probability of blue team success predicted for that number of sensor grids. The X axis indicates the blue team win percentage; the Y axis the lifetime cost of the sensor grids in the configuration; and the dots in the plot indicate the highest win percentage predicted for that number of sensor grids (from 1 to 10, going from left to right). The plot indicates that the incremental costs are buying a steady increase in win probability up to five, possibly six, sensor grids. At that point there is a sharp knee in the curve due to the probability getting sufficiently close to one. The added dollars are incapable of increasing the probability at the same rate beyond six sensor grids.

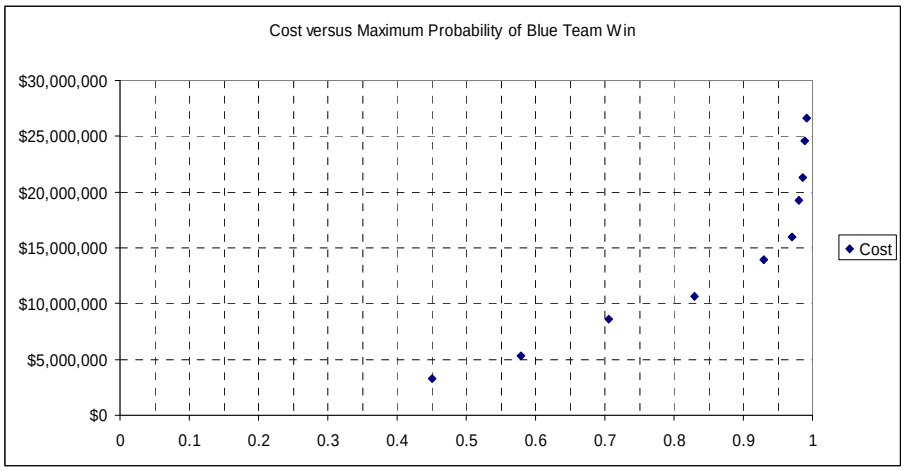

Figure 14. Cost of number of sensor grids versus maximum success probability

Summary and Conclusions

The DTA methodology is a structured, repeatable, and defensible analysis option that holds the promise of resolving the unsolvable, especially by untangling interdependencies. As was pointed out in the introduction, wicked problems tend to be heavily influenced by policy decisions. Therefore when developing the use case description of the system, policy constraints should be included as part of the stakeholder objectives and goals, or at least the results of the analysis should be vetted against the policy constraints.

Agile methods, as applied to software, can be extended to systems engineering. Similar processes have been successfully used by members of this team on large defense projects, and

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

small eCommerce projects. This paper demonstrates how they could be applied to the analysis of "wicked" problems using DTA. The key is to create the necessary and sufficient artifacts along the way, and to produce demonstrable products incrementally to shorten fielding time.

While the DTA will not always arrive at a satisfactory outcome, the results can be useful. The analysis may show that the original premise is flawed. For example, in the case study, the lifecycle cost to maintain the desired level of assured security may be too high. Such a conclusion would be valuable in itself. Significant resources and time could be saved that can be devoted instead to re-architecting the asset to create a truly secure, affordable solution.

As an analysis framework, DTA is particularly good at simplifying the interdependencies of complex systems. The authors believe that the transition between a SysML decomposition and a DSM dependency partitioning, and between that DSM partitioning and a simulation execution workflow, is a novel contribution of the work presented in this paper.

Future research opportunities include:

1. Definition of recommended roles and responsibilities for systems engineers applying agile principles;
2. List of best practices with explanation on how to implement this practice;
3. Mapping of this agile process into other frameworks used in systems engineering (DoDAF or MoDAF for instance); and
4. Tailoring guidance to make it agile enough to accommodate small and large projects while satisfying selected governance requirements.

Acknowledgements

The authors are indebted to several people at Sandia National Laboratories. Steve Tucker provided the impetus for the project by organizing the original research project. Michael Skroch provided needed management focus and drive. Michael McDonald provided numerous insights and guiding principles. Karen Page provided insights and analytic focus. Regina Griego provided systems engineering concepts, direction and input on content for the paper.

References

Balmelli, Laurent. (2007). "An Overview of the Systems Modeling Language for Products and Systems Development." *Journal of Object Technology* 6.6 (July-August), 149-177. Accessed on 10 June 2008 at http://www.jot.fm/issues/issue_2007_07/article2.

Barton, John, and Tim Haslett. (2007). "Analysis, Synthesis, Systems Thinking and the Scientific Method: Rediscovering the Importance of Open Systems." *Systems Research and Behavioral Science* 24, 143-155.

Cantor, Murray. (2003). "Organizing RUP SE Projects." *The Rational Edge*. July 2003.

Cho, Soo-Haeng. (2007). DSM Excel macro accessed on 18 March 2009 at http://grierson.nyscedii.buffalo.edu/dsmweb/dsm_tools/DSM-MIT-ver-1_9.zip.

Eppinger, S. D., D. E. Whitney, R. P. Smith, D. A. Gebala. (1994). "A model-based method for organizing tasks in product development." *Research in Engineering Design* 6.1, 1–13.

Linebarger, John M., Mark J. De Spain, Michael J. McDonald, Floyd W. Spencer, and Robert J. Cloutier. (2009). "The Design for Tractable Analysis (DTA) Framework: A Methodology for the Analysis and Simulation of Complex Systems." *International Journal of Decision Support Systems Technologies* 1.2 (April-June), 69-91.

Rational Software Company. (2001). "Rational Unified ProcessBest Practices for Software Development Teams." Rational Software White Paper, TP026B, Rev 11/01. http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf . Last accessed 3/18/2009.

Biography

Dr. Robert Cloutier is an Associate Professor of Systems Engineering in the School of Systems and Enterprises at Stevens Institute of Technology. He has over 20 years experience in systems engineering & architecting, software engineering, and project management in both commercial and defense industries. His interests include systems engineering patterns, systems architecting, model based systems engineering, and architecture management. Rob belongs to the International Council on Systems Engineering (INCOSE), IEEE and ACM. He received his Ph.D. in Systems Engineering from Stevens Institute of Technology, his M.B.A. from Eastern University, and his B.S. in Physical Science from the United States Naval Academy.

Mark J. De Spain has been an engineer at Sandia National Laboratories for over twenty years. He has worked in weapons components including sensing devices, firing sets and use control. Currently he is working as a weapon systems engineer. He has a BSME from Oregon State University and an MSEE from the University of Portland. He is currently enrolled in the doctoral systems engineering program at Stevens Institute of Technology. He is currently the president of the INCOSE Enchantment Chapter in New Mexico.

Dr. John M. Linebarger is a Principal Member of Technical Staff at Sandia National Laboratories. He has over 25 years of experience in the areas of distributed systems, collaborative virtual environments, semantic technologies, cognitive systems, statistical text analysis, and modeling and simulation. He has an MBA from New York University, an MS in Computer Science from the University of New Mexico, and a PhD in Computer Science from Lehigh University. Dr. Linebarger is a member of IEEE and the ACM.

Dr. Floyd W. Spencer has been a Distinguished Member of Technical Staff at Sandia National Laboratories for thirty years with experience in statistical analysis. Dr. Spencer retired in 2008.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.