

Rapid Concept Development of the Mission Space Architecture, Process Modeling, and Capability Analysis

David Flanigan, Bruce Schneider, Joseph Wolfrom
The Johns Hopkins University Applied Physics Laboratory
Laurel, MD 20723

Abstract—Early system engineering is not focused on the natural triangle of relationships between architecture, alternatives and performance. As a result, early understanding of conceptual trade space is focused on a sequential and independent effort when developing the system concept. This method is rife with inefficiency, broad and incorrect assumptions of the preceding step, ultimately leading to large amounts of rework and a non-unifying effort throughout the project team. This paper provides a means to show through an illustrative example that these three elements are interconnected and building them simultaneously improves the efficiency of concept development. The results of the paper highlight the parallel relationships between the elements that are usually developed in a serial process. This method provides the means to continuously build related architecture, alternatives, and performance elements together.

Index Terms— systems architecture; model based systems engineering; systems analysis

I. INTRODUCTION

Early system engineering is not focused on the natural triangle of relationships between architecture, alternatives and performance. As a result, early understanding of conceptual trade space is focused on a sequential and independent effort when developing the system concept. This method is rife with inefficiency, broad and incorrect assumptions of the preceding step, ultimately leading to large amounts of rework and a non-unifying effort throughout the project team. Often these team elements are geographically or organizationally separated, relying on either concept assumptions or low resolution communications to result in a less complete understanding of the details and nuances of the preceding artifact.

II. MOTIVATION

During early systems engineering development activities, engineering artifacts to describe the system concept are developed, which may include operational architectures, conceptual alternatives, and effectiveness simulation. Operational architectures may be used to describe the relevant actors, activities, and interfaces required to define the initial boundaries of the system concept. Conceptual alternatives may be used to describe the different solution types in terms of performance, cost, and technical risk. Effectiveness simulation considers both positive effects with other friendly nodes and potentially negative effects on mission success

based on the different operating environments and activities defined by the architecture. Separate groups and levels of expertise typically do these efforts, relying on different aspects of the customer to verify their products are satisfactory prior to progressing towards the next step.

Despite advances in software development using the incremental Agile approach, development of concepts is dominated by one-step-at-a-time engineering practices.

We also believe there are some indicators for when practitioners should NOT use this method. If any combination of these indicators are present in the project, we suggest that the concept development is not really needed *rapidly*. The indicators are: the sponsor cannot support the review cycle, the 80% answer is not sufficient, or expertise from the parent organization cannot be made available for an exclusive development effort. Another method may be more suitable for a project with these indicators.

III. LITERATURE REVIEW

Although our literature review discovered a sizeable set of papers on each of the components below, there are very few papers that provide guidance on how they should be integrated in a rapid concept development effort.

A. Operational Architecture

Friedenthal, Moore, and Steiner describe the basics of using the System Modeling Language (SysML) to model the operational architecture [1]. Maier and Rechtin submit that architecting deals with ill-structured situations, where neither goals nor means are known with much certainty, and that the architect seeks satisfactory and feasible problem-solution pairs [2]. De Neufville and Scholtes discuss the need for flexibility in engineering design due to uncertainties and the range of possibilities that may occur over the life of a system. These concepts are similar in nature to the case study presented in this paper, where the system under design is impacted by a great many unknowns [3].

B. Conceptual Architecture

1) Context Generation

This includes the environment for the system, adversaries or competitors, allies or partners, and the host platform(s). We started with the US Department of Defense Universal Joint Task List (UJTL) Joint Conditions reference from 2009 which

offers 2.1×10^{181} combinations of operating conditions for system designers to consider [4]. We followed the Joint Conditions structure and selected 16 conditions that would stress our alternatives in the mission space of our project.

2) Alternative Generation

A classic alternative generation method comes from Zwicky and his morphology box concept in which he developed the idea of listing the key alternative attributes in a row with their different means of accomplishing them in the column below each attribute [5]. He then challenged designers to consider all of the combinations as a way of expanding possibilities for new inventions. Hall uses the technique of morphological analysis to show a structure of systems engineering [6]. His cornucopia model of a narrowing engineering focus over time offers a useful definition of an effective method. Ritchey highlights the mechanics of using a software tool to aid morphology cross-consistency analysis to control combinational explosion [7]. We take his suggestion to include Bayesian Belief Networks into our method. Finally, Ullman offers some criteria for engineering decision support tools that is especially useful in our review phase [8].

C. Effectiveness Simulation

The literature on Capability Based Assessments (CBA) is studied since this is one of the primary tools that DoD has to develop initial requirements for their systems. The Department of Defense (DoD) utilizes CBA to assist with the initial analysis phase for new system acquisition [9]. Jones and Herslow describe the United States Air Force approach to capabilities-based planning and programming in a similar approach to the systems engineering method. They define the functions that describe the capabilities, the Concept of Operations (CONOPS) (that defines the flow of tasks to accomplish the capabilities), in order to evaluate the best course of action. Of importance are the assessment questions that are asked: (1) How good are we? (2) Do we have enough of whatever it takes? (3) What is the impact? [10]. Davis describes a process to conduct capabilities based planning that develops the scenarios, identifies alternatives and metrics, and assesses alternatives to select the best capability option. His process implies a linear flow to address a single concept and does not explicitly account for alternatives needing to address multiple problems that may be found within a System of System (SoS) problem [11]. Fry and DeLaurentis implement a single adjacency matrix to account for the system-system interactions within the SoS [12]. We may leverage this approach in order to catalog the types of interactions that must be represented in the simulation environment.

IV. CASE STUDY

For our case study, we looked at applying our method to mission space of a unmanned aerial vehicle (UAV) sensor trade study. An example of a “Sense and Reactive Intelligence Surveillance and Reconnaissance (SR-ISR)” concept is used to explore the conceptual trade space for an Unmanned Aerial

Vehicle (UAV) that reports adversary target detections while avoiding hostile air defense activities during a mission.

A. Problem Statement

a) *Objective Function*- the SR-ISR system will maximize detection of adversary targets while minimizing exposure to the threat surface-to-air missile (SAM) sensors and weapons.

b) *Decision Variables*- the initial UAV decision variables were: sensor range, processing delay time, UAV speed, and UAV alternative type. These are developed into a more detailed set in the section D below.

c) *Key Constraints*- the UAV flies a certain flight path based on the alternative selected. The effectiveness simulation is a deterministic process that evaluates the number of detections that the UAV makes on the red SAM based on the configuration versus the number of detects of the red SAM on the UAV based on position in the scenario. The UAV sensor performance is constrained by weather degradation. The UAV communication performance is constrained by weather degradation and adversary jamming of its communications receiver

B. Method overview

Our method builds architecture, conceptual models, and simulations simultaneously to support a rapid conceptual development task. The method forms the development into two or more rounds of effort. Each round is a pre-defined duration event (e.g. 2 hours, days, or weeks) that is guaranteed to produce some output relevant to the project. The round is supported by three stakeholder types: Requirements, Alternatives, and Test. Figure 1 shows their relationships.

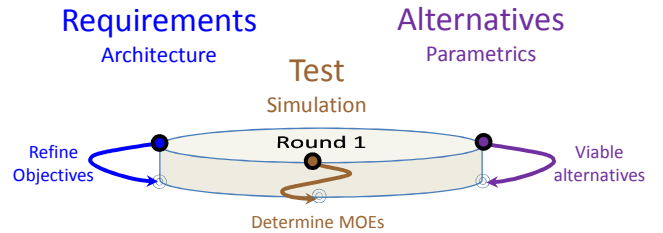


Fig. 1. Simultaneous engineering round

The round starts with each of the stakeholders asking six basic questions of each other. These questions request information from the other stakeholders that is required to complete the round of concept development. At this point in the round, each stakeholder is searching for constraints in the other stakeholder's areas as shown in Fig. 2. This acknowledgement of the interdependencies of the three stakeholders sets the stage for efficient work within the round.

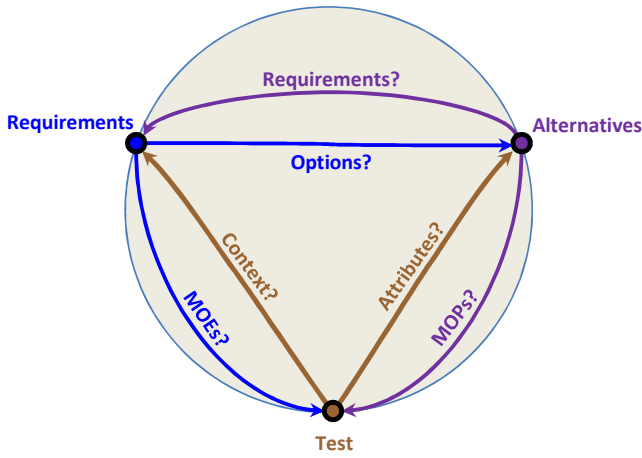


Fig. 2. Beginning questions for the round (top of the cylinder).

The stakeholders commit to answering the questions by the end of the round. The round's fixed duration determines the time allotted and sets the expected abstraction level of detail for the results. We will look at the interior processes of the round later in this paper. The end of the round shows each of the stakeholders providing the required answers given the round's duration, as shown in Fig. 3.

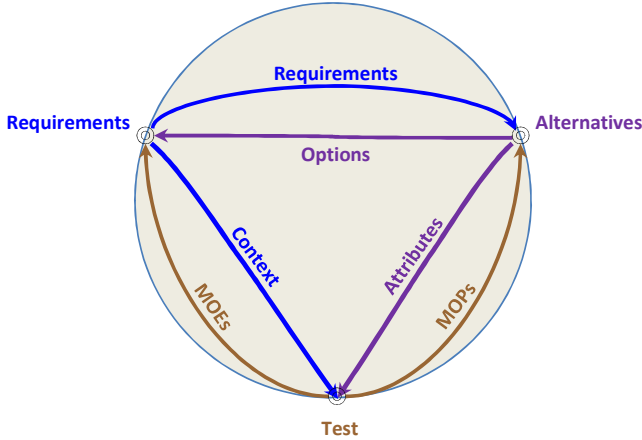


Fig. 3. Ending answers for the round (bottom of the cylinder).

After each round, a review is held with the sponsor. The review is designed to communicate the team's current status, obtain the sponsor's updated preferences, and prioritize the efforts for the next round. Each round becomes more in depth and less in width than the previous round, as shown in Fig. 4.

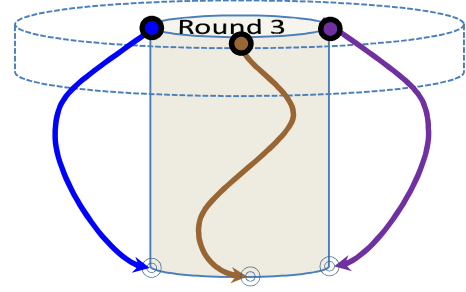


Fig. 4. Round 3 is has less solution-width and more solution-depth than Rounds 1 or 2 before it.

Next, we look at the interior processes of each round that provide rapid concept development. Each round is broken into four arcs. These are the formal exchange of information between the three stakeholder types. Our preferred method is to conduct each arc meeting with support from a common on-line model repository. The arcs are listed in below in table I along with a sampling of questions, estimates, and product types for each stakeholder.

TABLE I. MILESTONES WITHIN EACH ROUND (ARCS)

Arcs		Stakeholder results (sampling)		
#	Termination Criteria	Requirements	Alternatives	Test
1/4	Prioritize questions	Nodes?	Communications reliability?	Sensor range?
2/4	Provide estimates	Air, Ground	30%, 60%, 100%	20, 40, 60
3/4	Review applied estimates	Alternative architectures	Context performance impact	Effectiveness Curves
4/4	Record results (SysML)	Activity Diagram	Block Description Diagram	Parametric Diagram

C. Operational Architecture Application

For this case study, the conceptual design team is presented with the challenge of allocating functionality. In the simplest terms, the system functions can be depicted as consisting of the following four functions: Observe, Orient, Decide, and Act (OODA). The conceptual design team must determine how to allocate these functions. One alternative is to allocate all four functions to the UAV as depicted in Fig. 5.

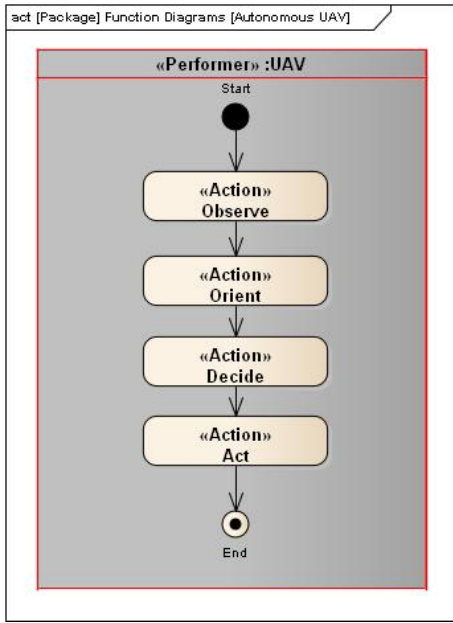


Fig. 5. Alternative 1: Autonomous UAV.

Another alternative considered in this case study is to allocate some of the functions to the UAV (Observe and Act), and some of the functions to the Ground Station (Orient and Decide). This alternative is depicted in Fig. 6.

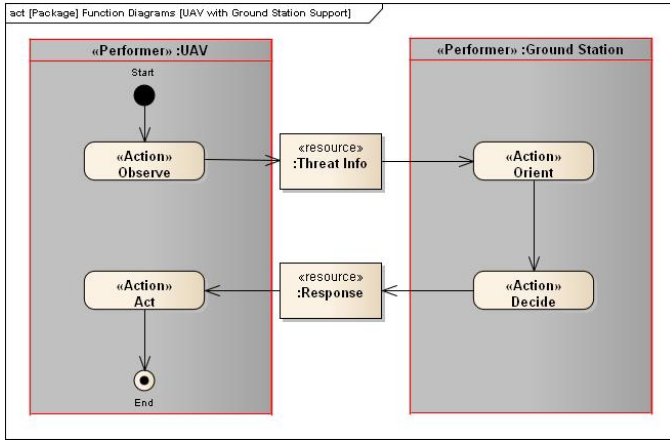


Fig. 6. Alternative 2: UAV with Ground Station Support.

The challenge for the conceptual design team is to develop a methodology to evaluate these alternatives. However, more information is needed pertaining to the environment in which the system is expected to perform. For example, one might expect that Alternative 1 is a better solution in a communications-constrained environment, due to the message exchanges that are required in Alternative 2. However, one might expect that Alternative 2 is a better solution in a cost-constrained environment due to the increased complexity of the UAV internal processors required by Alternative 1. Thus the conceptual design team must perform a trade study to evaluate the alternatives. However the alternatives are impacted by

several unknown factors, (such as the level of communications jamming). The methodology of performing the conceptual alternative evaluation is discussed in the next section.

D. Conceptual Alternative Application

The results of the Arc 1 meeting set the tasks for the alternative stakeholders. Their objective was to provide the rest of the team with a small set of alternatives that were both internally-consistent and representatively-sampled across the mission space. To support Arc 2, the alternative stakeholders completed initial estimates for five interlocking models.

Cost model. A simple cost model was developed to estimate the acquisition and annual operational costs for each alternative. The primary goal of the cost model was to make the same financial constraints apply to all alternatives. Although there could be better performance available at higher budget levels, forcing each alternative to "live within its means" allowed the team to compare alternative's performance for dollar expended.

Conditions model. A set of relevant conditions were generated for the analysis. These represent the environment that the alternative concept will operate in. The condition model had three constraints. Each condition must stress the alternative in some way. Each condition must choose an internally consistent set of decision variables to describe its properties. Finally, the set of conditions had to provide the team with a diversified sampling of condition cases.

The team developed four conditions with two or three properties each. Weather Degradation defined the reduction in sensor detection performance due to weather. The levels were 0 and 50% reduction. Jamming defined the reduction in UAV communications due to adversary electronic attack. The levels were 30, 60, 100% communications reliability. UAV platform speed defined the ground speed of the UAV. The levels were 100, 200, and 300 knots. The platform flight path defined how directly the UAV would fly towards the suspected SAM sites. The levels were aggressive (straight over suspected area), semi-conservative (slight off-set), and conservative (completely standoff). The summary of the team's key environmental conditions variables is show below in TABLE II.

TABLE II. SR-ISR CONDITIONS MODEL

Condition Variables	Parametrics
Weather Degradation	-0% , -50% sensor detection probability / glimpse
UAV communication reliability	30% (contested jamming environment), 60% (semi-permissive environment), 100% (permissive environment)
UAV platform speed (knots)	100, 200, 300
UAV platform flight path	1 (aggressive), 2 (semi-conservative), 3 (conservative)

Alternative design. The alternatives stakeholder developed some alternatives that fit within four key constraints. Each alternative had to achieve performance about the objective. Each alternative had to choose an internally consistent set of decision variables to describe its properties.

Each alternative had to fit within the stated constraints of the analysis. Finally, the set of alternatives had to provide the team with a diversified sampling of variable combinations.

The team developed six alternatives with two properties: Sensor Range and Processor Delay. Sensor Range defined the maximum range of the UAV's sensor to detect the adversary targets. The levels were 20, 40, and 60 nm. Processor Delay defined where the target identification function resided according to the functional architecture (Fig. 5 and 6). The levels were Remote (Ground Station) or Autonomous (onboard UAV). These settings translated into a UAV sensor processing delay of 100 or 50 seconds respectively. Alternatives were identified by these properties (e.g. the UAV named *SR.60-Proc.Auto* had a 60nm sensor range and on-board target ID processing)

The conditions model and alternative designs were then combined into a condition-alternative matrix. This combined matrix was examined for missing condition-alternative pairings that were not duplicated or dominated by another pairing within the matrix. The summary of the alternatives' key properties are show below in TABLE III.

TABLE III. SR-ISR ALTERNATIVES

Alternative Name	UAV sensor range (NM)	UAV sensor processing delay (sec)
<i>SR.20-Proc.Remote</i>	20	100 [Ground Station]
<i>SR.40-Proc.Remote</i>	40	100
<i>SR.60-Proc.Remote</i>	60	100
<i>SR.20-Proc.Auto</i>	20	50 [UAV autonomous]
<i>SR.40-Proc.Auto</i>	40	50
<i>SR.60-Proc.Auto</i>	60	50

Performance modeling (sensors, com links, costs). The combination of condition variables and alternative properties generated a 324 performance modeling matrix. This matrix formed the basis of the simulation run matrix discussed in section E. The alternative stakeholders also provided the team with some generic models for relating each object's operation or reception with a mathematical relationship to the component variables. These models provide the simulation with a method to relate internal processing speed and accuracy with external condition impacts to sensor and communication performance. The pseudo code is given below:

for each time step

Detect_SAM = success iff:

Range.Target <= [UAV.Sensor.Range
– (UAV.speed * Processor.DetectionSpeed)]
* WeatherDegradation.Sensor
* Comms.success

Comms.success = 1 iff:

Comms.required = 1
AND WeatherDegradation.Comms = 1
AND Jamming.UAV.comms.receiver = 1
OR Comms required = 0

otherwise Comms.success = 0

where

Range.Target is the slant range from the UAV to the target (Nm)

UAV.Sensor.Range is the maximum range that the alternative's sensor can detect a target (Nm)

UAV.speed is the UAV ground speed (Nm/sec)

Processor.DetectionSpeed is the time from the UAV's sensor observation to the declaration of a target (sec)

WeatherDegradation.Sensor is a random variable set to 0 if the weather prevents the sensor from observing the target. {0, 1}

Comms.required is a binary set to 1 if the alternative requires an off-board connection to declare a target track

WeatherDegradation.Comms is a random variable set to 0 if the weather prevents UAV comms receiver from receiving the ground station processed track. {0, 1}

Jamming.UAV.comms.receiver is a random variable set to 0 if the UAV comms receiver does not hear the ground station processed track {30, 60, 100% comms reliability}

Internal impacts to performance (condition effects).

The alternative stakeholders examined each of the alternative's functions under each condition. They developed an estimated internal performance change matrix based on the impact that each condition would have on each function. For example, hot temperature conditions might slow down the processing speed. These matrices would be applied at simulator run-time to account for major internal performance impacts caused by external conditions. Note that these internal impacts are in addition to the external impacts caused by conditions (e.g. rain degrading sensor range)

To support Arc 3, the alternative stakeholders reviewed the other team member's analysis and provided top three challenges for both Requirements and Test stakeholders.

To complete Arc 4 actions, the alternative stakeholders updated their portion of the team's online model repository with final data from the other team member Arc 3 challenges and lessons learned.

E. Effectiveness Simulation Application

The effectiveness simulation was written in MATLAB, a general purpose modeling language. The intent of using a general tool was to represent the activities and interactions of the ISR system as described by the operational architecture. The scope of the tool is lower fidelity to ensure a rapid prototype of actions, interfaces, and metrics were developed to facilitate a quick turn-around to the other team members.

The simulation was developed from the architecture activity diagrams that show the roles, activities, and who they interface with. To augment the simulation, a sequence diagram was also used to understand the order of interfaces. Some of the questions that were raised were how to model the differences between the ISR alternatives, how can the main activities of

detection, processing, and understanding be modeled, and how to model the environmental effect on both red and blue platforms, such as weather and jamming. A series of short meetings were held to discuss these topics, with parts of the team dispersing to develop the models and then reconvening to verify the accuracy of the modeling interpretation. After several attempts, it became apparent that architecture of the effectiveness simulation proved most beneficial in conveying the details and logic flows to the other parts of the team. Figure 7 provides an example of the architecture of the simulation used to promote information exchange within the team.

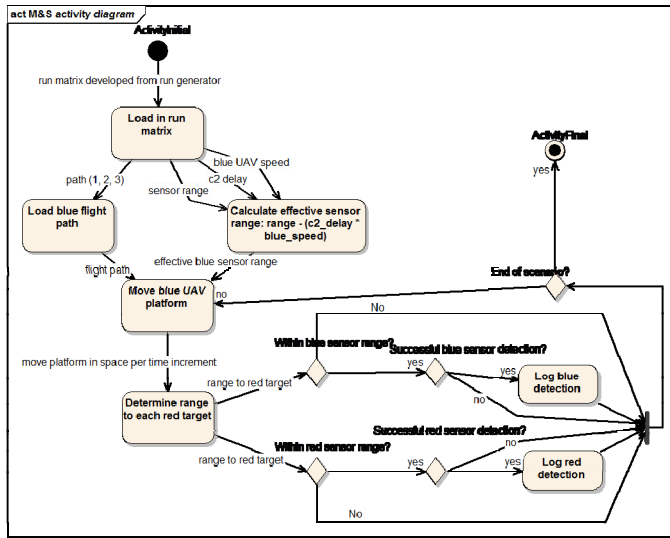


Fig. 7. Simulation architecture – activity diagram.

The initial simulation runs considered a wider tradespace in terms of platform performance: speed, sensor range, command and control delays, weather effects, and jamming effects. The parameter settings were intentionally set wide to capture a variety of configurations that may be initially deemed infeasible. By capturing more configurations, we avoid any pre-selection or filtering bias on ISR configurations without the analysis. We believe this provides a more comprehensive and unbiased view of the potential tradespace.

The initial simulation results identified the critical areas of ISR utility, with the main metrics being blue detection, red detection, and a "threshold of understanding" that was developed by achieving a defined amount of blue detections. This was represented by the uncertainty of information as inversely proportional to the range of the red target, with a cumulative level of understanding gained by the sensor and the command center. Once this threshold was achieved, the amount of exposure to the red threat was also calculated, to provide a sense of risk / reward for the different ISR alternatives. Figure 8 provides an example of such an output from the simulation. Each of the three ISR configurations is represented in a different color and symbol. The x-axis represents how long the UAV took within the scenario in order

to achieve the knowledge threshold, and the y-axis represents the number of red detections during that time, representing the amount of risk that the UAV takes during the duration of the mission. The desired region of the plot is in the bottom left, where knowledge is quickly gained at a low level of risk. The next desirable region is the bottom right, where knowledge is slowly gained, but still at a lower risk. The least desirable region is the top right, where it requires a long amount of time to gain knowledge and puts the UAV at a large exposure risk. The top left quadrant is a high risk, high reward zone for quick knowledge with large amounts of risk involved.

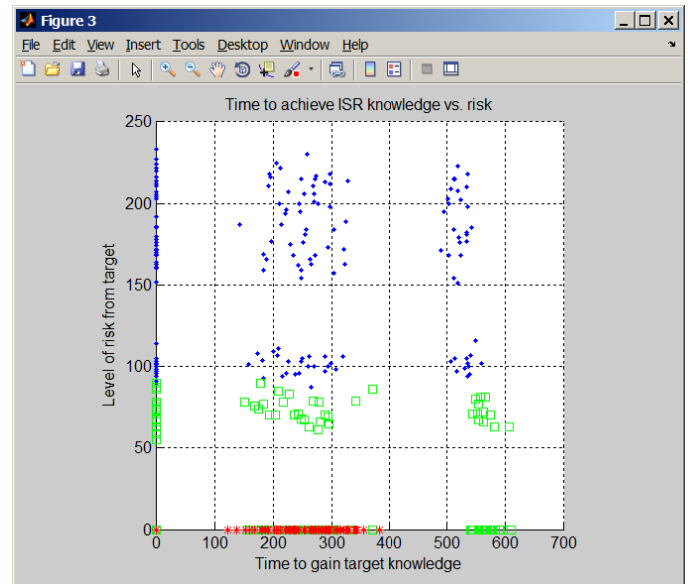


Fig. 8. ISR Simulation Results.

Configuration 1 is the blue dot that represents the most direct route towards the target set, and thus takes the most risk to red threat exposure. Dependent on the ISR capability description, it will span between 150-550 seconds to acquire the needed knowledge for the scenario, but also expose itself to a greater amount of risk, and resides in the two least desirable quadrants of the risk/reward graph. Configuration 2 is the green square that takes a slight offset and a more conservative flight path around the target. It resides in the bottom two quadrants of the risk/reward graph, but suffers from a range of long ISR dwell times to achieve the knowledge threshold. Configuration 3 is the red asterisk that follows the most conservative flight path that tries to avoid the majority of the threat areas, and resides in the most desirable quadrant of the risk/reward graph. Based on these results, configurations 1 and 2 had similar success rates (66%) to achieve the detection thresholds, while configuration 3 had a lower success rate (44%), likely due to the conservative flight route around the target to emphasize survivability.

After consultation with the team, selected variables were chosen to improve the fidelity of variables and their effect on the ISR mission and associated metrics.

F. Round Completion Review

We then conducted an iterative review among the requirements, alternatives, and test (simulation) team members in order to evaluate the next steps and record the team current belief in the problem's description and potential solutions.

Our preferred method was to use the ACCORD [8] software from Robust Decision Inc. This allowed the team to simultaneously enter both their satisfaction and uncertainty for each alternative and criterion. Each stakeholder brought data for the other stakeholders to consider. The submitted data was entered into the software under the appropriate stakeholder (e.g. Under the Requirements stakeholder, scenario likelihood data was entered. Under the Alternative stakeholder, cost data was entered. Under the Test stakeholder, simulation results of MOE was entered. From the submitted data starting point, each stakeholder recorded how accurately the other stakeholder's data described the final system. In this way, the review process captures data products and the team's current belief in the products as shown in Fig. 9. In future rounds, the software allows the stakeholders to update their beliefs and resample the consensus of the team. The spread of team's opinion allowed the software to recommend "what to do next." Depending on the team's marks, new alternatives should be developed, weak alternatives should be deleted, and more knowledge on specific aspects should be researched. The software's Bayesian belief map method focused on what actions will increase consensus the most in the next round.

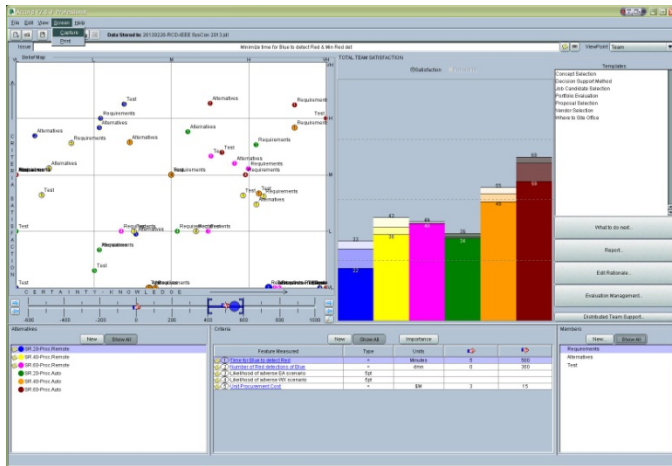


Fig. 9. Round Completion Review (ACCORD software).

As part of the review analysis, ACCORD recommended that in the next round the team focus on building consensus on Time for Blue to detect Red for *SR.20-Proc.Auto* alternative and increasing team knowledge about the likelihood of adverse EA scenario for *SR.60-Proc.Auto* alternative. These are the actions that have the most leverage in narrowing the choices in the next round.

Table III provides a summary of the types of products that are expected from each round, with emphasis on the increasing fidelity of detail based on the interactions within the team.

TABLE IV. FIDELITY OF ARTIFACTS DEVELOPED WITHIN EACH ROUND

Phase	Requirements	Alternatives	Test
1	Development of an objectives sketch	Commencement of functional analysis	Initial mission measures of effectiveness (MOE) and measures of performance (MOP) development
2	Development of an objectives hierarchy	Creation of Zwicky box results	Creation of MOE / MOP distributions for differing alternatives
3	Quantified objectives	Block Definition Diagram (bdd) of each alternative	Optimized alternatives MOP values based on MOE results

V. CONCLUSION

The Rapid Concept Development method applies the lessons learned from Agile software development to rapid concept development. We enhanced this method by employing standardized modeling repository techniques (SysML), deliberative alternative generation (architecture-based allocation assignments and morphology), condition-based simulation (custom MATLAB coding), and Bayesian belief maps-supported reviews (ACCORD).

ACKNOWLEDGMENT

We'd like to thank our colleague, Leo Mayoral for his excellent ideas on early solutions exclusions and when *not* to apply a rapid method.

REFERENCES

- [1] S. Friedenthal, A. Moore and R. Steiner, A Practical Guide to SysML, The Systems Modeling Language, New York: Morgan Kaufmann, 2012.
- [2] M. Maier and E. Rechtin, The Art of Systems Architecting, New York: CRC Press, 2009.
- [3] R. de Neufville and S. Scholtes, Flexibility in Engineering Design, Cambridge, MA: MIT Press, 2011.
- [4] Department Of Defense, *Universal Joint Task List*, Washington, DC: Department of Defense, 2009.
- [5] F. Zwicky and A. G. Wilson, New methods of thought and procedure, Springer, 1967.
- [6] A. D. Hall, "Three-Dimensional Morphology of Systems Engineering," *Systems Science and Cybernetics, IEEE Transactions*, vol. 5, no. 2, pp. 156-160, 1969.
- [7] T. Ritchey, "Problem structuring using computer-aided morphological analysis," *Journal of the Operational Research Society*, vol. 57, pp. 792-801, 2006.
- [8] D. G. Ullman, "The ideal engineering decision support system," 2002. [Online]. [Accessed 10 March 2009].
- [9] Joint Chiefs of Staff, *Capabilities-Based Assessment (CBA) User's Guide Version 2*, Washington, DC: US Department of Defense, 2006.
- [10] J. Jones and R. Herslow, *The United States Air Force*

- Approach to Capabilities-Based Planning & Programming (CBP&P), Part 1: Planning*, Washington, DC: Air Force Studies and Analysis Agency (AFSAA), 2005.
- [11] P. K. Davis, *Analytic Architecture for Capabilities-Based Planning, Mission-System Analysis, and Transformation*, Santa Monica, CA: Rand Corp, 2002.
- [12] D. N. Fry and D. A. DeLaurentis, "Measuring Net-Centricity," in *6th International Conference on System of Systems Engineering (SoSE)*, Albuquerque, NM, June 27-30, 2011.