75 Fifth Street NW, Suite 213
Atlanta, GA 30308, USA
Voice: +1-404-592-6897
Web: www.InterCAX.com
E-mail: info@intercax.com

Using ParaMagic and SysML in Combination with UPDM

Dr. Dirk Zwemer
InterCAX LLC

Abstract

An approach is proposed that uses UPDM for “big picture” specification of defense systems in combination with SysML for parametric simulation of specific scenarios and use cases. We consider an example of a MagicDraw UPDM model of a reconnaissance-and-response system for hostile missile launches, with two SysML models that calculate system performance against specific kinds of threats. Our example also includes the use of Cameo Requirements Plus for scenario storyboarding at the use case development stage. The MagicDraw ParaMagic plug-in connects the SysML models with Microsoft Excel®, MATLAB® and Mathematica® to initialize, execute, report and visualize the simulation results.

Introduction

The relationship between UPDM (Unified Profile for DoDAF/MoDAF) and SysML (Systems Engineering Modeling Language) is an active question today. On one side, UPDM is simply a domain-specific variant of SysML, with all the same capabilities, but optimized for defense applications. On the other side, the objectives of building UPDM and SysML models are frequently quite different, suggesting both have a role to play. The purpose of this application note is to describe one approach to using MagicDraw’s SysML, UPDM and ParaMagic plug-ins, in combination with Cameo Requirements Plus, to efficiently develop complex defense systems.

Table 1 Comparison of UPDM and SysML/ParaMagic

UPDM	SysML/ParaMagic
A structured series of diagrams following the DoDAF/MoDAF model	No specific diagrams required
Describes architecture of defense system responding to a threat	Describes complete system, including threat
Focuses on communications and organizational relationships	Focuses on hierarchical, flow and parametric relationships
Architecture-oriented, descriptive	Performance-oriented, simulations
Qualitative	Quantitative
Includes all use cases and scenarios	Focuses on single scenarios

Table 1 reflects the author's opinions, that UPDM is used to specify systems in a precisely defined way for a particular audience, while SysML is used more generally to explore a system's potential performance, particularly when a parametric solving tool such as ParaMagic is incorporated. The last item originates in the current state of technology for parametric solvers, not SysML itself. Parametric-based simulations tend to assume a particular sequence of events (a single scenario) because sequences with branching points (potential for multiple scenarios) are not easy to solve. To treat different scenarios, we create different SysML models. These may have many common elements, but different parametric networks of constraints.

NoMagic's Cameo Requirements Plus (Req+) offers an important new piece to the system development process. In addition to a requirements manager, Req+ provides story-boarding capability. Specific scenarios can be organized and illustrated, to assist stakeholders in understanding the system and generating meaningful requirements.

Magic Draw UPDM, SysML/ParaMagic, and Cameo Req+ can work together iteratively

- Use Cameo Req+ to generate requirements, use cases and scenarios
- Use MagicDraw UPDM to design overall architecture
- Use MagicDraw/SysML to evaluate architecture under specific scenarios

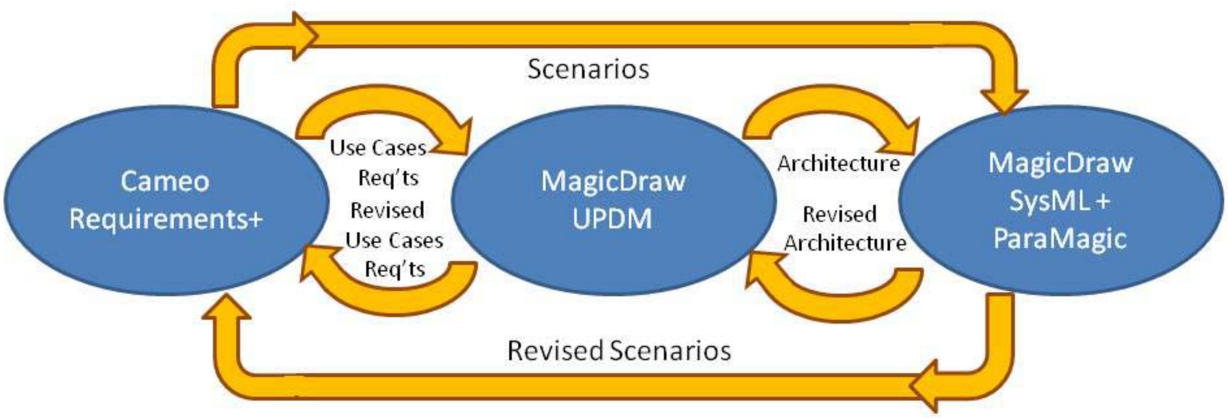


Figure 1 Proposed System Development Process

The System

Our system development process starts with a threat scenario. An enemy mobile missile launcher leaves concealment, travels to a launch site, fires a missile, and returns to concealment. The defender has two responses.

Scenario 1. A squadron of LittleEye UAVs (unmanned aerial vehicles) scans the battlespace for launchers. If one is located, a Special Forces team is dispatched to "paint" the target with a laser designator and an F-18 is assigned to destroy the target, preferably before a missile can be fired, but at least before the launcher can return to concealment.

Scenario 2. If a missile is fired, an aerial sensor vehicle (GlobalHawk) detects the launch and trajectory and alerts multiple batteries of Patriot counter-missiles. If the missile comes within range of one of the batteries, the battery will attempt to shoot the missile down.

As a first step, Cameo Req+ can be used to generate a sequence of events (Figure 2) in one scenario, for example, a successful mission to destroy the launcher. A storyboard (Figure 3) illustrates the events for rapid understanding. Similar storyboards can be generated other scenarios, including the Patriot system.

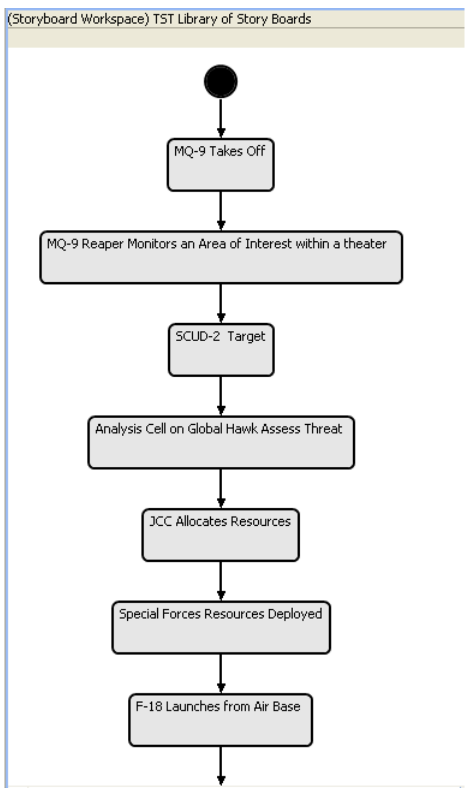


Figure 2 Scenario sequence of events (partial)

Figure 3 Scenario storyboard (partial)

A UPDM model is formulated to represent the defense system and the time-sensitive-threat (TST) it addresses. The OV-1 and OV-2 diagrams shown in Figures 4 and 5 offer "big picture" overviews of the system components and the communications required between them.

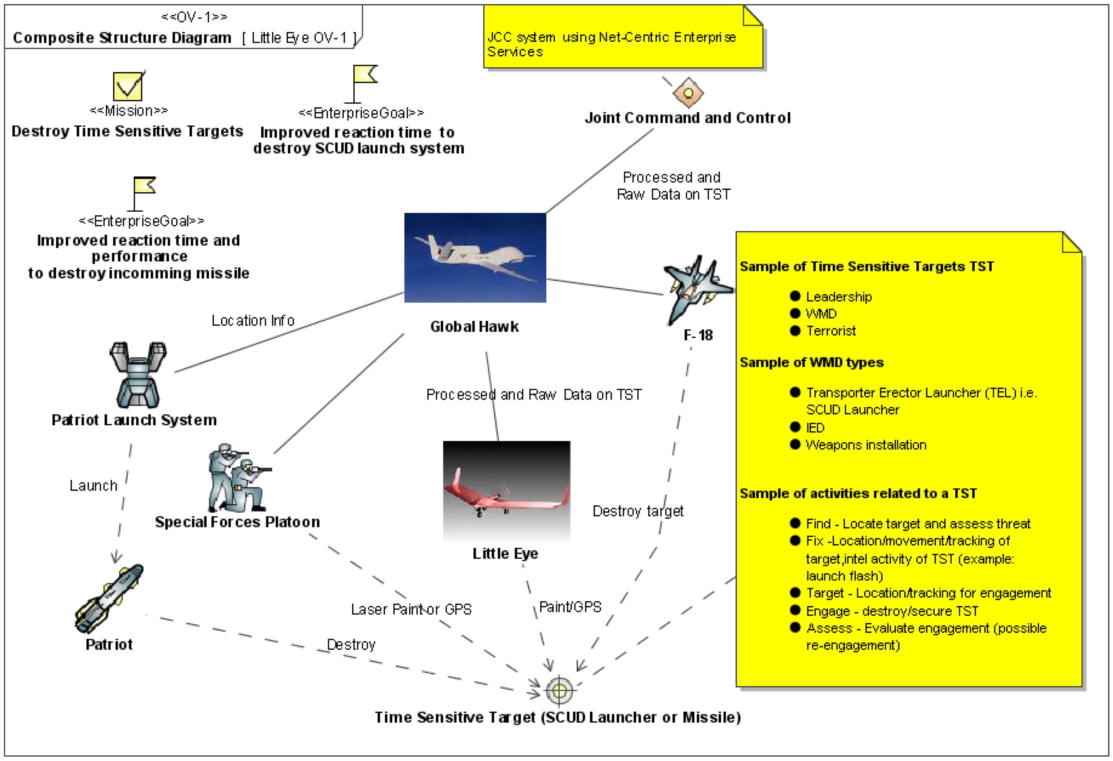


Figure 4 UPDM OV-1 diagram

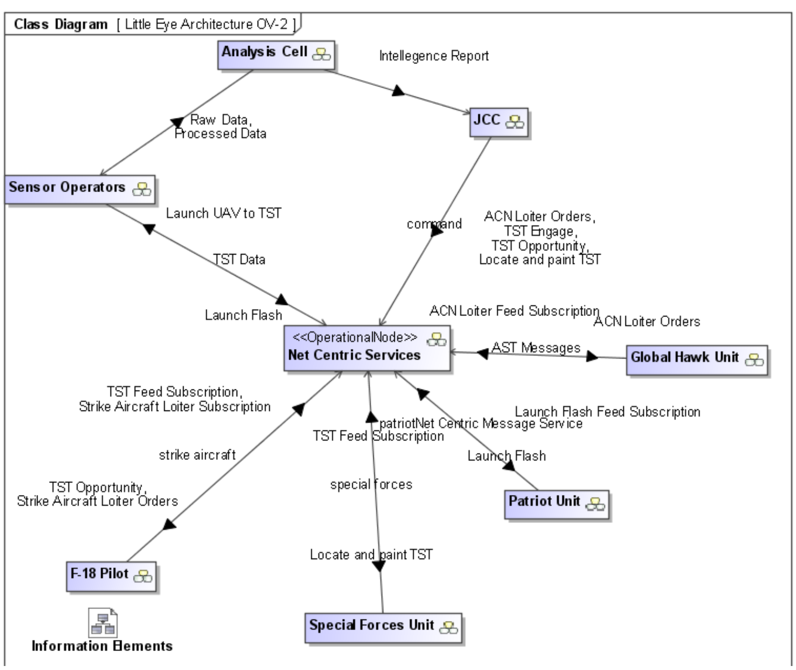


Figure 5 UPDM OV-2 diagram

The SysML Models

We have created two independent models of the UPDM for Scenario 1 and Scenario 2. In each case, we will start with an Activity diagram and a Block Definition diagram.

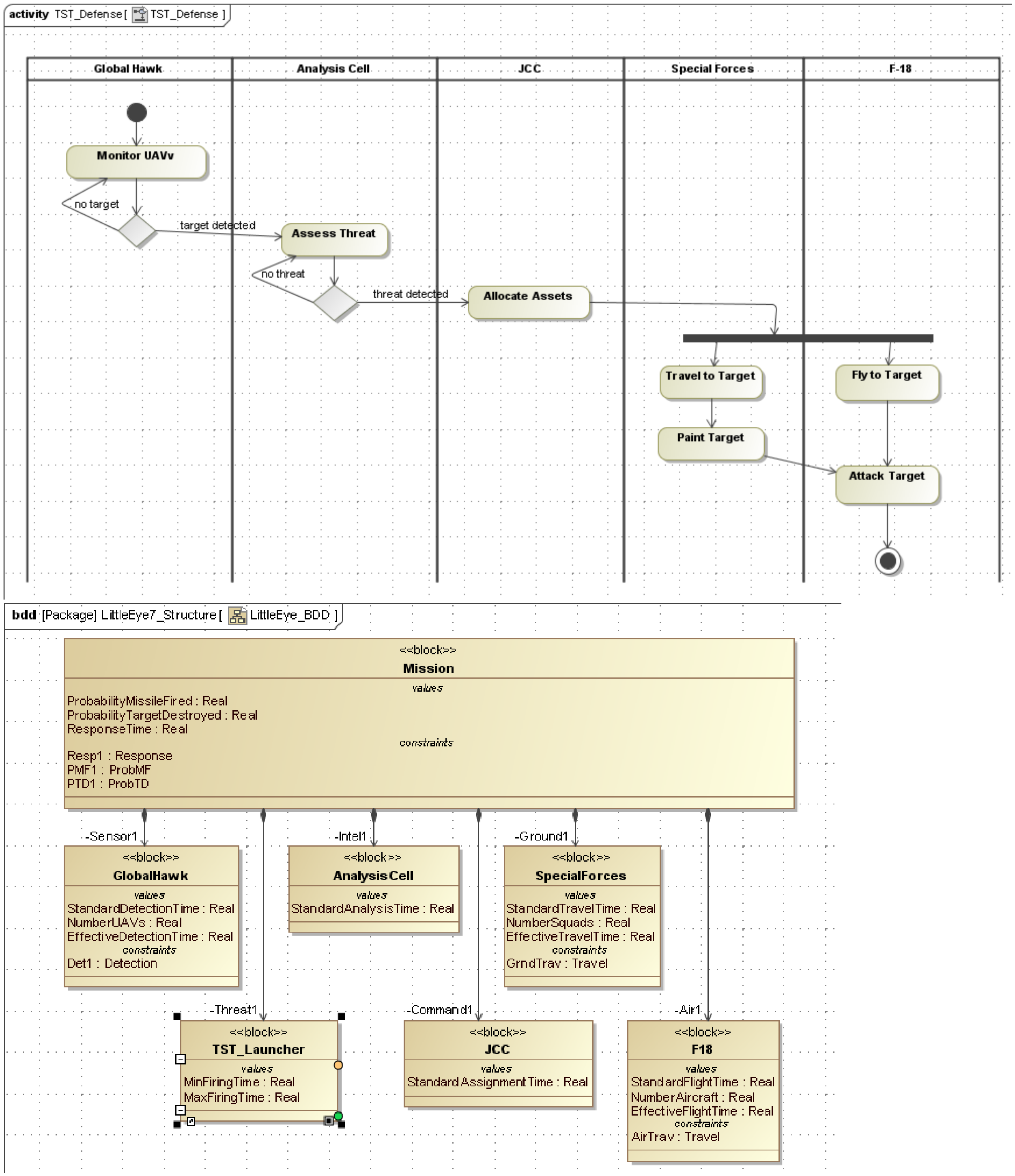


Figure 6 Scenario 1 SysML Diagrams

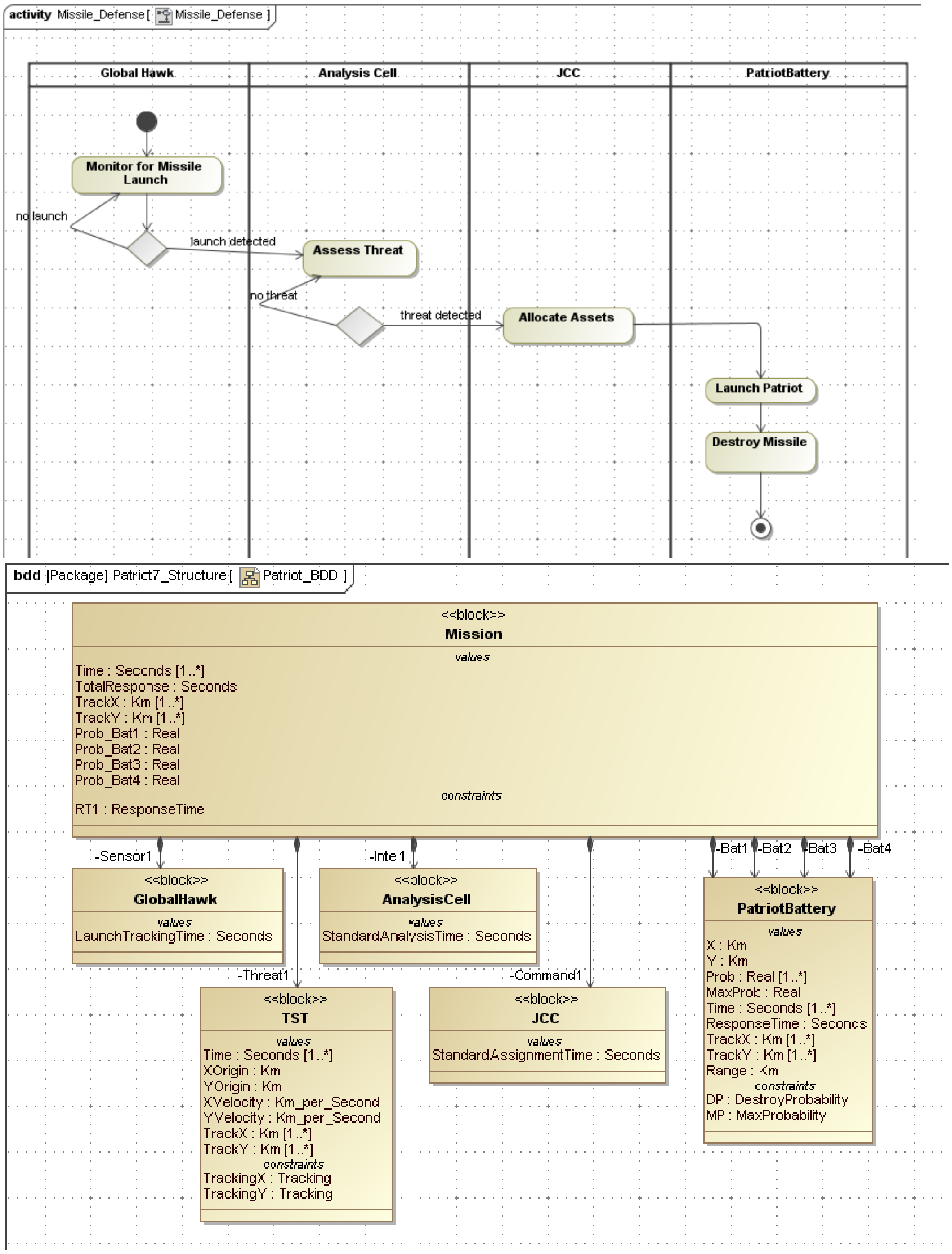


Figure 7 Scenario 2 SysML Diagrams

The activity diagrams in Figures 6 and 7 should correspond to OV-5 diagrams from the UPDM model. The block definition diagrams (BDD) do not correspond directly to anything in UPDM, although they do incorporate many of the elements from the OV-1 diagram in Figure 4. These BDDs are specifically designed for the parametric simulation to be performed with that SysML model. These simulations are described in the following sections.

The SysML Parametric Model – Scenario 1

The objective of this simulation is to calculate 1) the probability the missile launcher will fire a missile before being destroyed and 2) the probability the missile launcher will be destroyed. To build this simulation, we begin by assuming a threat profile:

1. The launcher leaves concealment, reaches the launch site, and prepares for launch with a minimum time MinFiringTime (units are not explicit in this model, assume hours)
2. There is a uniform probability of missile launch over a time interval (MaxFiringTime – MinFiringTime).
3. The launcher returns to concealment immediately after launch with a return time equal to MinFiringTime.

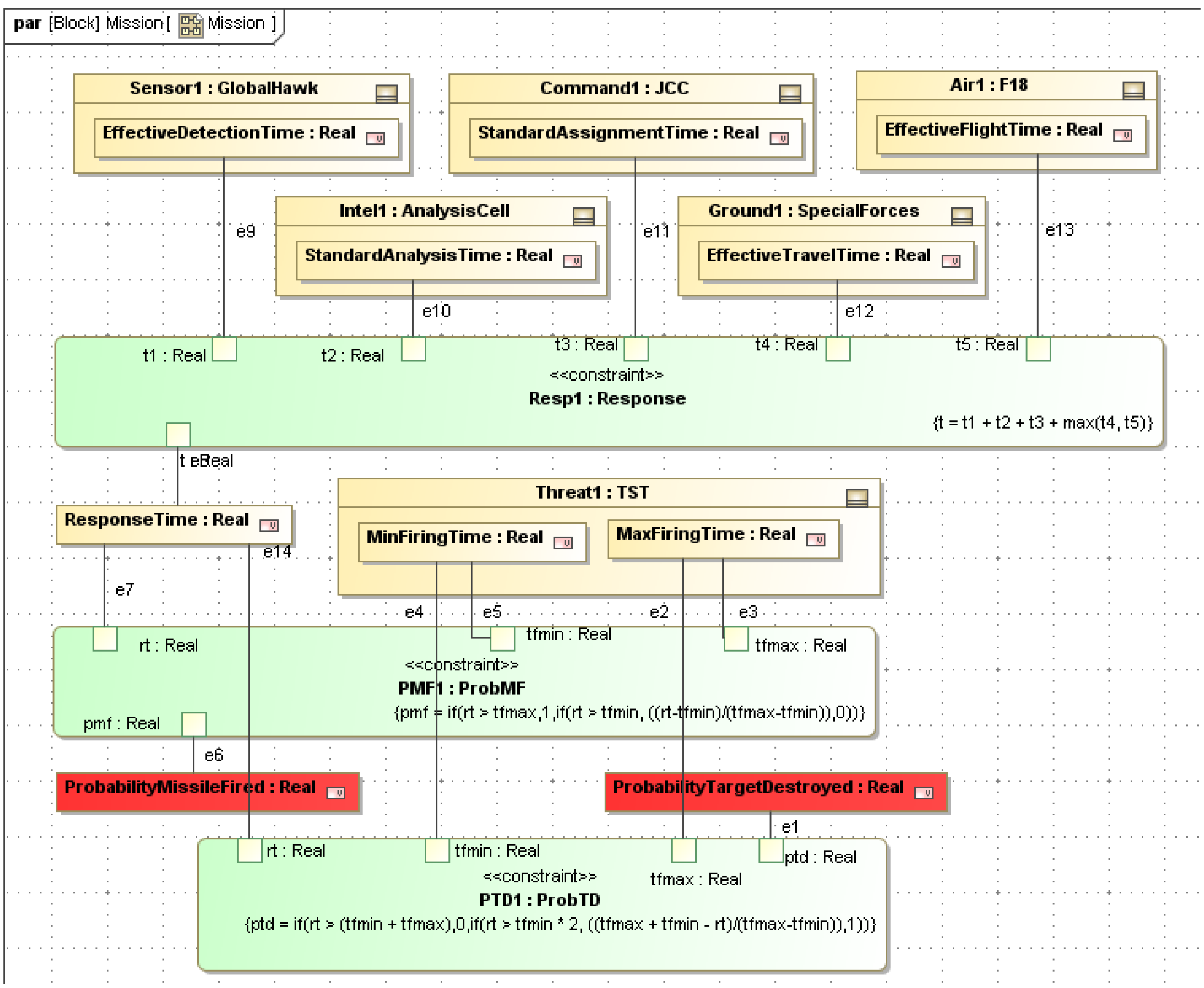


Figure 8 Parametric Diagram – Scenario 1

The success of the defense, in this model, depends on response time, reaching the launcher before it fires a missile or returns to concealment. The response time is the sum of

1. UAV detection time, which depends on the number of UAVs deployed
2. Intelligence analysis plus command decision time
3. Special Forces and F-18 travel time to launch site, which depends on the number of each deployed.

Performance is not only a function of system architecture; it is also a function of resources.

The high level parametric diagram for this model is shown in Figure 8. Constraint properties are shown in green. The primary outputs of the simulation model are shown in red. The blocks for GlobalHawk, SpecialForces and F-18 include additional constraints (not shown) that calculate response time as a function of number of units deployed.

Executing the SysML Parametric Model – Scenario 1

We can use ParaMagic to calculate ProbabilityMissileFired (before launcher destruction) and ProbabilityTargetDestroyed (before launcher returns to concealment) given the necessary input parameters. Figure 9a shows one parameter set before solving. Figure 9b shows the same browser after solution. The ProbabilityMissileFired is 1 (100%) and the ProbabilityTargetDestroyed is 0.494, less than 50%.

Assuming these figures are lower than desired, we can try adding additional resources to the problem. Quick analysis of 9b shows that average travel time to the launch site is much shorter for the F-18 than ground forces, so increasing the number of available fighters doesn't affect total response time. Figure 9c doubles the number of UAVs and the number of Special Forces squads. ProbabilityMissileFired drops to 0.78 and the ProbabilityTargetDestroyed is 100%. An alternate approach is to look for ways to decrease the characteristic response time of each element in the chain.

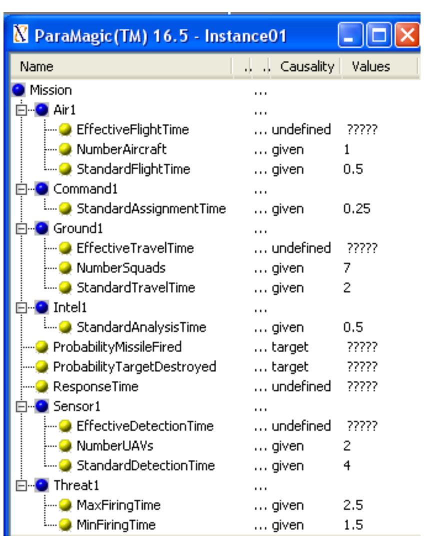


Figure 9a ParaMagic Browser before solution

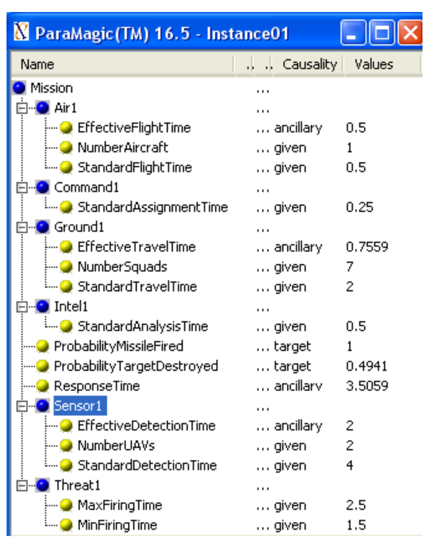


Figure 9b ParaMagic Browser, first solution

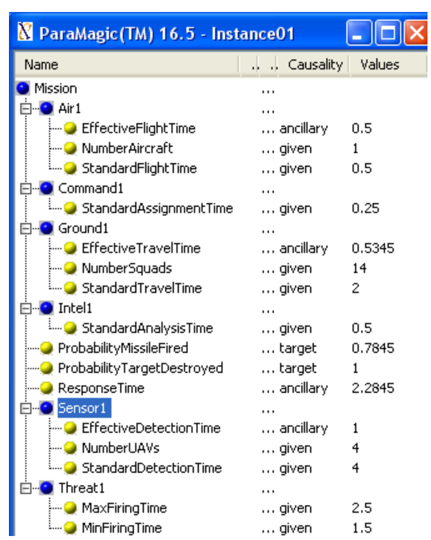


Figure 9c ParaMagic Browser, revised solution

The SysML Parametric Model – Scenario 2

The objective of the second simulation is to calculate that a Patriot counter-missile will intercept an enemy missile after firing. We will use two MATLAB functions and several equation-based constraints solved by Mathematica.

1. The MATLAB function tracking.m calculates the course of the threat missile over time, based on its initial firing point and velocity. The output is a list of missile locations at 100 second time intervals.
2. The MATLAB function destroy.m calculates the probability of destroying the threat with a Patriot, based on the location of the Patriot battery and the threat tracking results. The output is a list of missile intercept probabilities at 100 second time intervals.
3. Other constraints calculate the defense response time, which is the sum of detection time by the GlobalHawk sensor platform, analysis time by the intelligence team, and asset allocation time by the command function. The probability of intercepting the missile early in its flight, before the defense can respond, is zero.
4. The model finally extracts the maximum probability over time for each of four patriot missile batteries to intercept.

Figure 10 shows the parametric diagram for the intercept calculation by a Patriot missile battery. The MATLAB function is incorporated into the model as the constraint block DP:DestroyProbability (green, on left) where the inputs are the battery location (X and Y, in kilometers on an arbitrary grid), the threat tracking data (TrackX and TrackY, on the same grid, at 100 second intervals), a characteristic range parameter for the Patriot (Range, used in calculating the intercept probability), the Response Time of the defense, and Time (a list that sets the time scale for the calculations {100,200,300,400} in our example). A second constraint block, MP:MaxProbability, accepts the output of the destroy.m, a list of intercept probabilities at each of the time points in Time[1..*], and reports the maximum value, the maximum intercept probability for that Patriot battery. In our model, the same calculation is done for each of four batteries.

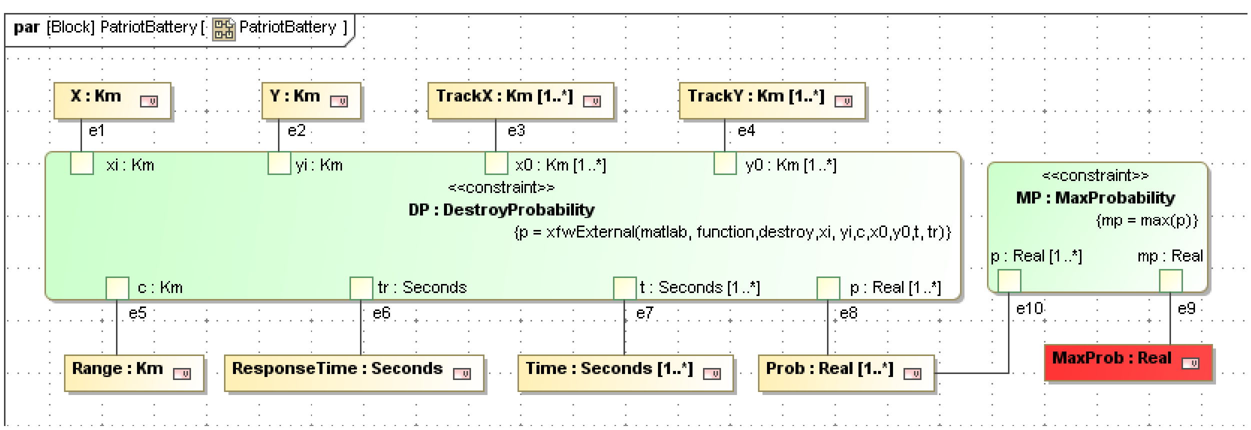


Figure 10 Parametric Diagram – Scenario 2

Executing the SysML Parametric Model – Scenario 2

We use Microsoft Excel as the I/O mechanism for this parametric model, providing a simple interface for entering the input parameters for a particular trial and for reporting the output parameters, including graphing of the results. The sequence of action for the simulation is

1. Enter input values in Patriot_v2.xlsx spreadsheet.
2. Upload inputs to SysML model instance.
3. Run parametric calculation
4. Download outputs to spreadsheet.
5. Use graphing functions to help interpret results.

The ParaMagic tool is used to execute steps 2, 3 and 4.

Figure 11 shows worksheet 1 of this spreadsheet. The user enters the initial location and velocity of the threat, the locations of the four Patriot batteries, and the Effective Kill Distance , the Range parameter in the intercept calculation in Figure 11. At the bottom, the maximum intercept probabilities for each of the four batteries has been calculated, with the best chance for Battery4.

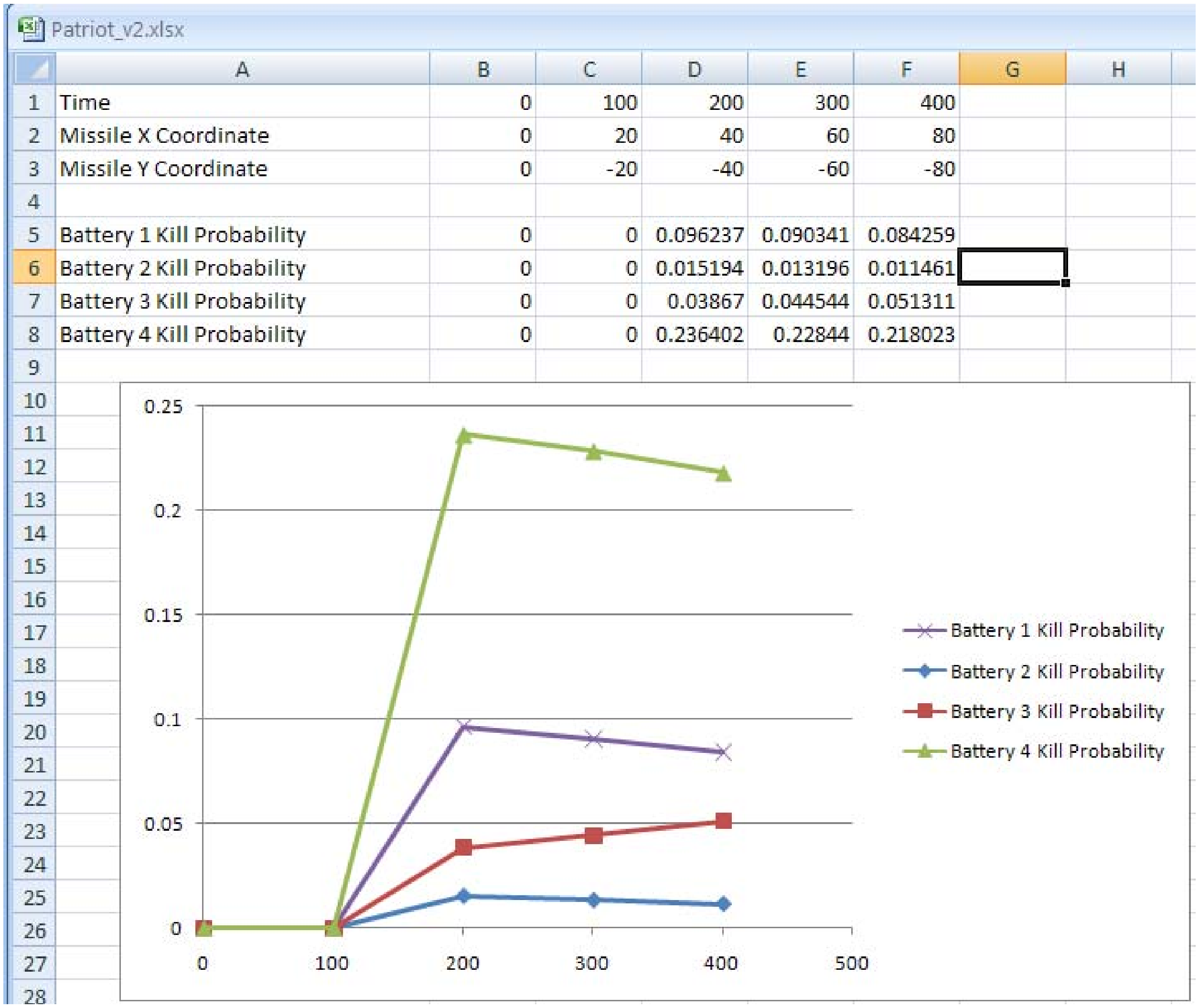
Patriot_v2.xlsx

	A	B
1		
2		
3	Time-Sensitive Threat	
4	X Coordinate (km)	0
5	Y Coordinate (km)	0
6	X Velocity (km/sec)	0.2
7	Y Velocity (km/sec)	-0.2
8		
9	Patriot Missile Batteries	
10	Effective Kill Distance (km)	200
11	Battery1	
12	X Coordinate (km)	200
13	Y Coordinate (km)	400
14	Battery2	
15	X Coordinate (km)	-600
16	Y Coordinate (km)	500
17	Battery3	
18	X Coordinate (km)	500
19	Y Coordinate (km)	-500
20	Battery4	
21	X Coordinate (km)	-200
22	Y Coordinate (km)	-200
23		
24	Max Missile Kill Probability	
25	Battery1	0.0962
26	Battery2	0.0152
27	Battery3	0.0513
28	Battery4	0.2364

Figure 11 Input/Output spreadsheet for Scenario 2

Figure 12 demonstrates how to take advantage of Excel for analyzing and graphing the SysML simulation results. In the top rows of the second worksheet of Patriot_v2.xlsx, we have downloaded the time-dependent results from the SysML parametric simulation, including the threat track and the probability that each battery can intercept, given the relative locations of battery and threat missile at that time point. In our model, the intercept probability drops exponentially with distance. Probabilities before the defense response time (detection, analysis, decision) are zero.

Results in our simplified model are easy to interpret. The intercept probability for Battery3 increases with time, because the missile is headed “southeast” and is closing the gap. However, Battery4 is much closer to the initial launch site, so has higher intercept probability in the initial stages. If we extended the simulation to longer times, we might see a higher ultimate probability for Battery3, if the missile stays on the same course.


Patriot_v2.xlsx

	A	B	C	D	E	F	G	H
1	Time	0	100	200	300	400		
2	Missile X Coordinate	0	20	40	60	80		
3	Missile Y Coordinate	0	-20	-40	-60	-80		
4								
5	Battery 1 Kill Probability	0	0	0.096237	0.090341	0.084259		
6	Battery 2 Kill Probability	0	0	0.015194	0.013196	0.011461		
7	Battery 3 Kill Probability	0	0	0.03867	0.044544	0.051311		
8	Battery 4 Kill Probability	0	0	0.236402	0.22844	0.218023		

Figure 12 Results Analysis of Scenario 2

Discussion

We've attempted to demonstrate how using SysML parametrics to integrate legacy models can combine the power of proven models and specialized simulation tools with the high-level visualization and collaboration advantages of model-based systems engineering. Several directions for future development appear.

1. It would be better for UPDM and SysML models to be integrated into single projects, to minimize duplication. Guidelines for creating SysML diagrams from UPDM objects would be useful.
2. Interfaces to more simulation tools and models would be valuable, including C++ and Java code modules, discrete event simulators, CAE tools like FEM and CFD, and many others.
3. Mechanisms for efficiently exploring the operations space (trade studies, design of experiments) are needed.
4. Activity diagrams are central to UPDM models. SysML parametric models that can use the information captured in behavior diagrams would simplify UPDM-SysML integration.