75 Fifth Street NW, Suite 312
Atlanta, GA 30308, USA
voice: +1-404-592-6897
web: www.InterCAX.com
email: info@InterCAX.com

Modeling *The Martian*

Dr. Dirk Zwemer | InterCAX LLC | dirk@intercax.com | Oct 16, 2015

Abstract

Two scenarios from *The Martian*, an adventure story describing the challenges of a NASA astronaut inadvertently marooned on Mars, are modeled in SysML, with special attention to capturing the detailed parametrics from the novel. The first section explores the agronomics of growing food on a planet without normal air, water, soil or plant life. The second sections describes transportation planning for extended travel without a highly developed road infrastructure. The SysML parametric models are intended to offer some guidance on adding these features to complex MBSE systems, in addition to whatever entertainment value may result for the terminally engineering-minded.

Introduction

The Martian, a novel by Andrew Weir (Broadway Books, 2014), is a great read. Part of its appeal to engineers is that the astronaut hero “shows” his calculations on how to survive his many challenges alone on the planet Mars. The realism and resourcefulness of his work make his fate that much more real to the reader and heighten the emotional impact of the story.

Unfortunately, the engineers in the novel aren’t shown as using model-based systems engineering (MBSE). In the interest of future astronauts, I will try to capture some of the lessons learned on the Ares 3 mission in the form of SysML parametric models (while trying to minimize spoilers for those yet to enjoy the book or the movie).

Two scenarios, captured as use cases in Figure 1, will be modeled. In the first, the astronaut plans to extend the food stocks available by growing food inside his Martian base camp. In the second, he needs to travel a long distance across the Martian landscape, carrying his energy supply, his life support equipment, and the food and water required to reach his destination. The figures use MagicDraw (No Magic Inc.) to create and ParaMagic (InterCAX) and Mathematica (Wolfram Research) to solve the model, but the same model will also be made available for download from the InterCAX website (www.intercax.com) in several forms, as they are completed and verified:

- Enterprise Architect (Sparx Systems), solved with Solvea (InterCAX)
- IBM Rational Rhapsody (IBM), solved with Melody (InterCAX)
- Integrity Modeler (PTC), solved with ParaSolver (InterCAX)
- MagicDraw 18.0 (No Magic), solved with ParaMagic (InterCAX)

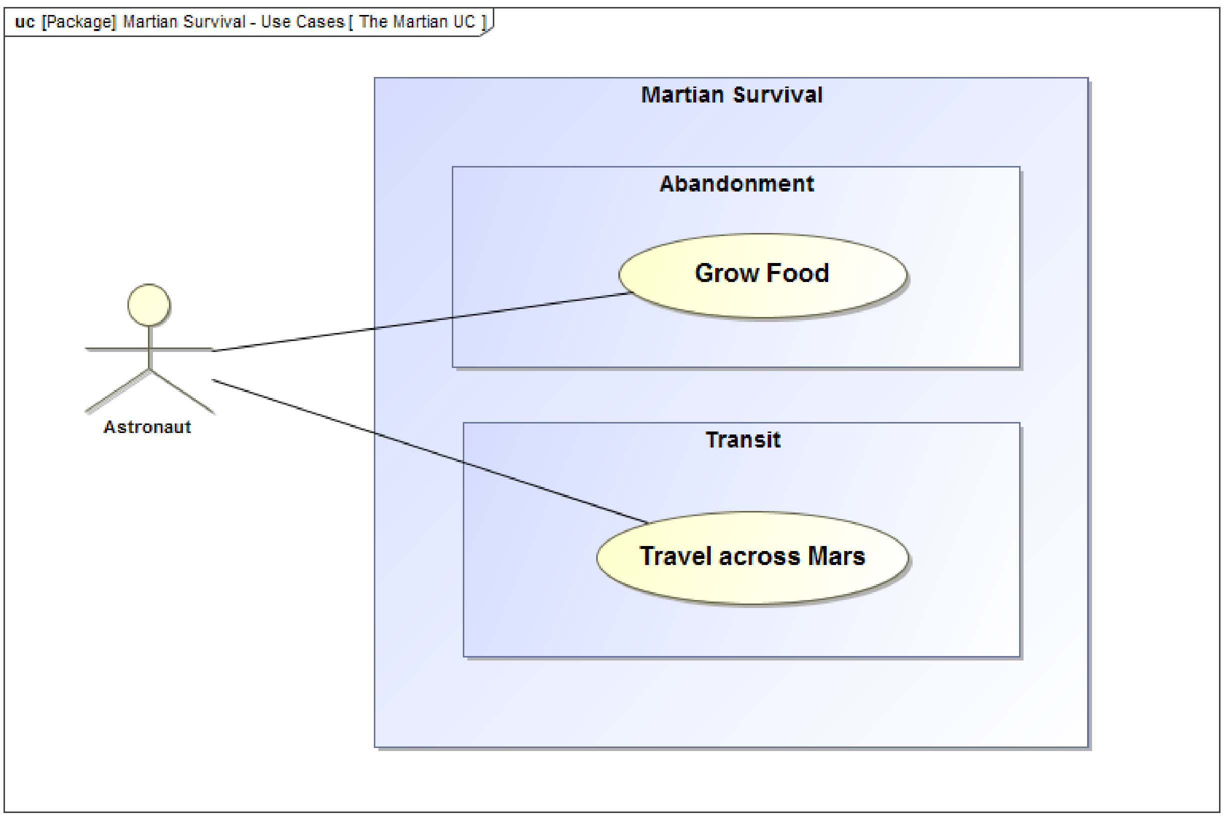

Figure 1 Astronaut objectives under two sceanrios

Martian Survival - Abandonment

The first scenario is set at the Acidalia Planitia landing base of the Ares 3 mission. The BDD in Figure 2 shows that the base consists of the main habitat, two mobile Mars rovers (each with a pop up shelter), and the landing stage of the MAV (Mars Ascent Vehicle). It also includes the food stocks expected to be used under the original mission plan.

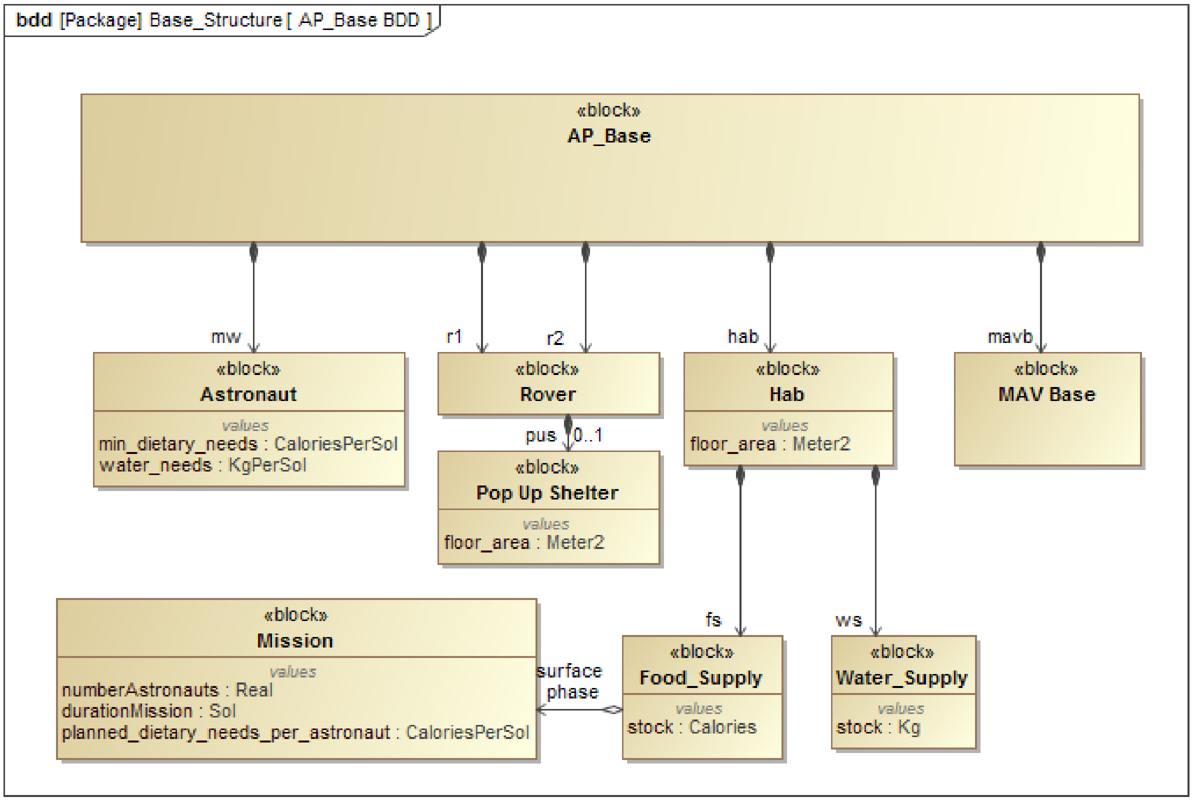

Figure 2 AP Base structure

The astronaut, a botanist among other talents, wants to grow enough additional food to survive until rescue. His analysis falls into four parts:

1. How much planting area can he create inside the habitable elements of the base?
2. How much water does the soil require for plant growth?
3. How much food (he plants potatoes) can he grow per sol (Martian day)?
4. How long will his extended food supply last?

I have structured these questions as four separate analysis blocks containing parametric models, as shown in Figure 3. These analysis blocks reference the AP_Base block, as well as each other, to access the necessary values for their calculations, without requiring changes to the properties or parametrics of the structure blocks.

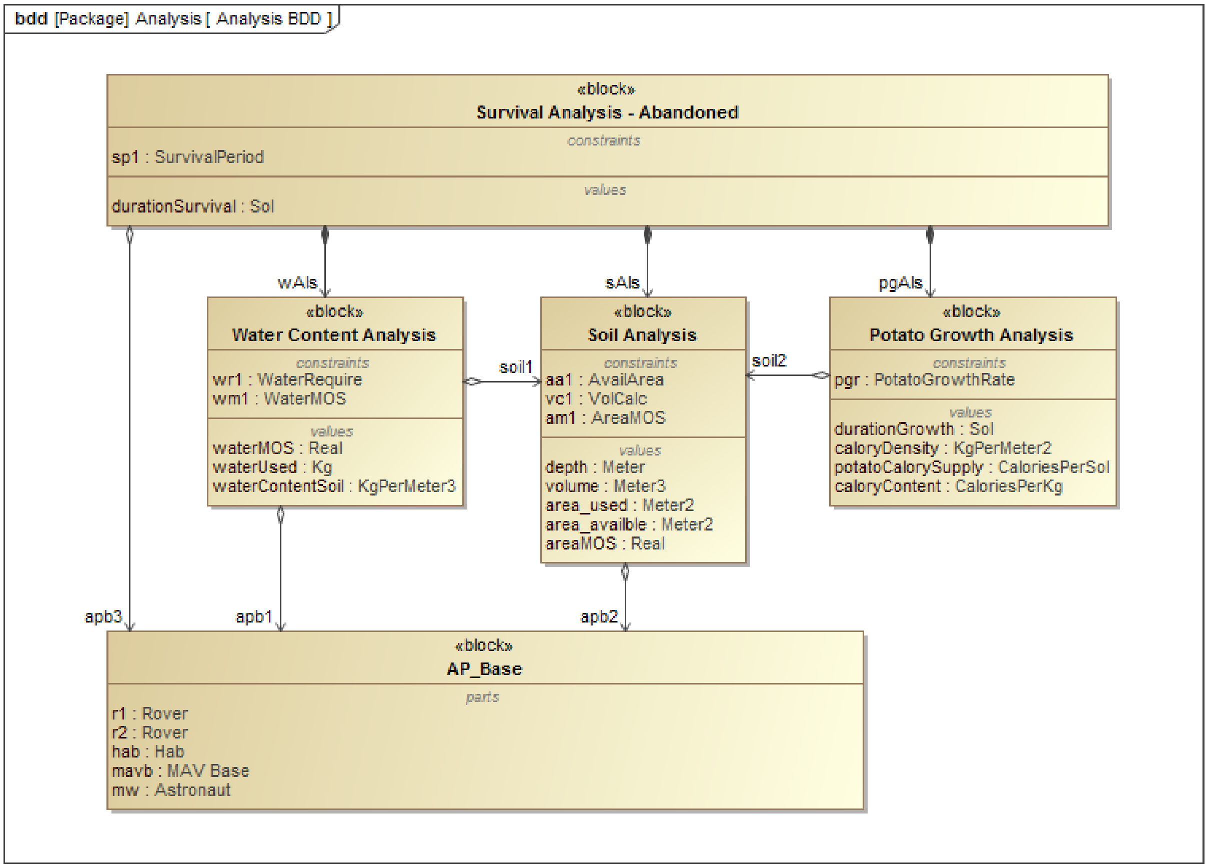

Figure 3 Analysis structure, Martian Survival – Abandonment scenario

The only parametrics embedded within the AP_Base blocks are for the food supply, shown in Figure 4.

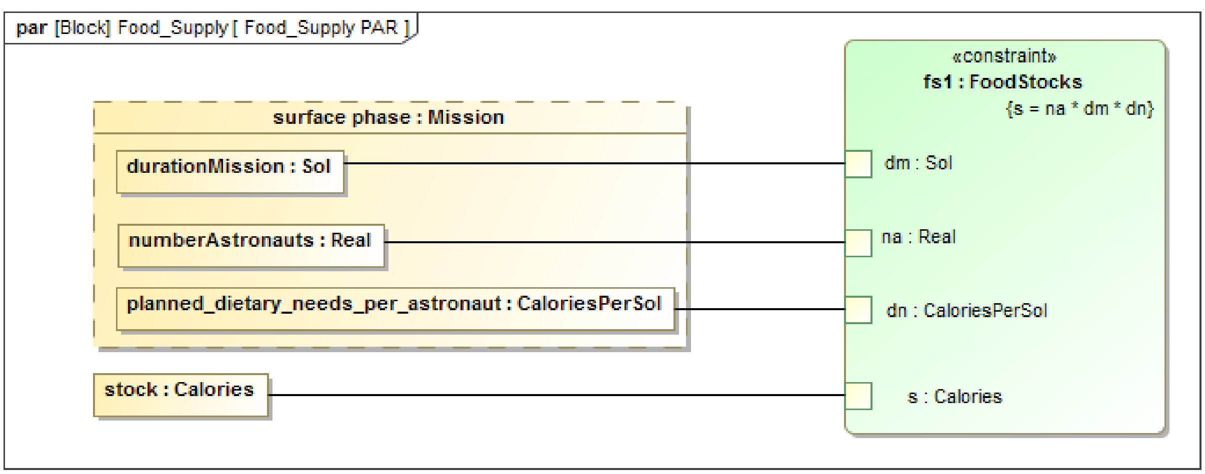

Figure 4 Food Supply PAR, calculating base food stocks based on the original mission plan (e.g. 6 people for 30 sols). The <equal> stereotype labels on the binding connectors have been elided from all diagrams for clarity.

Soil Analysis PAR (Figure 5)

This model compares the arable area in square meters created inside the habitat and pop up shelters against available area, ultimately calculating an area MOS (margin of safety) value. The MOS is positive as long as the area planned to be used does not exceed the area available. It also calculates the volume of soil to be created, which is used by Water Analysis.

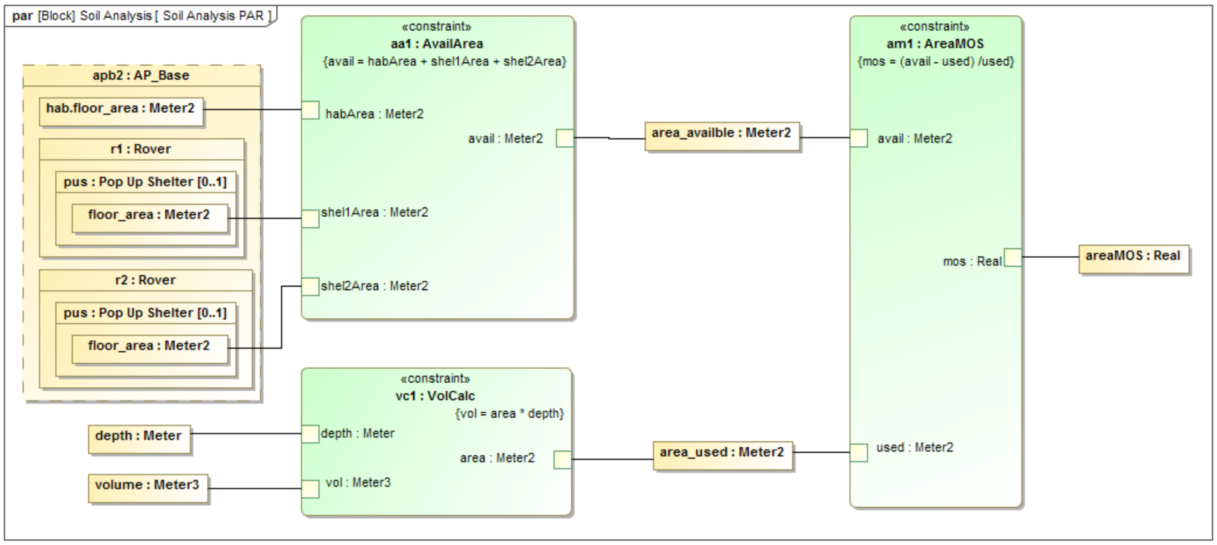

Figure 5 Soil Analysis PAR

Water Content Analysis PAR (Figure 6)

This part of the model calculates the water required to make the soil fertile with the stock available at the base, and calculates another MOS.

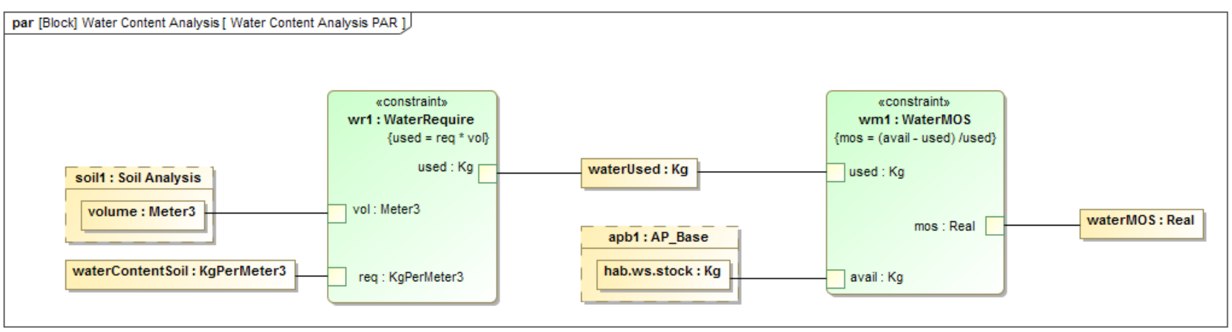

Figure 6 Water Analysis PAR

Potato Growth Analysis PAR (Figure 7)

This calculates the average number of calories generated from potatoes per sol. This ignores the initial lag while the potatoes are growing, since the final value of interest, the number of additional days of food, is not concerned with when the potatoes are eaten, as long as the first crop comes in before the initial food stock is consumed.

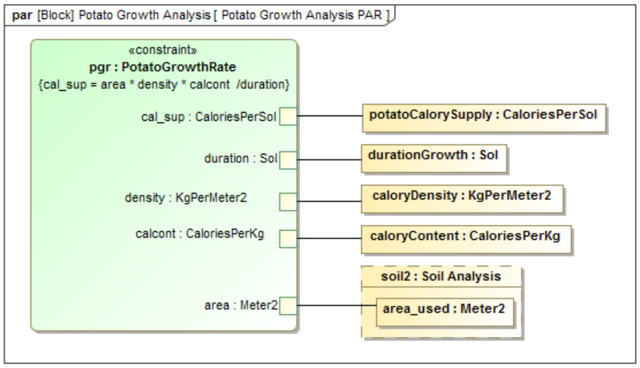

Figure 7 Potato Growth Analysis PAR

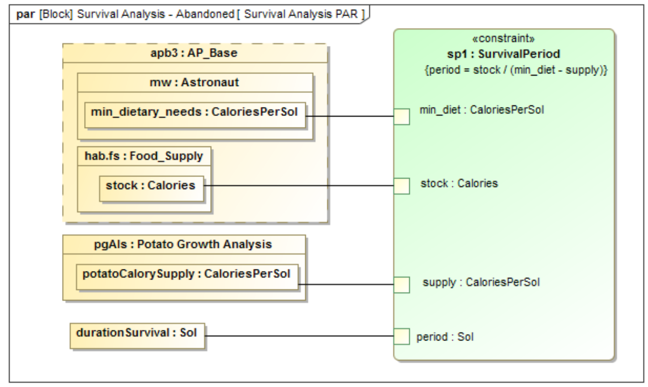

Figure 8 Survival Analysis PAR

Survival Analysis PAR (Figure 8)

This final section of the model determines the total number of sol that the minimum dietary need of the astronaut can be met between the original food stock and the additional food grown. Note that the period increases if more food can be grown, but the analysis becomes invalid (period becomes negative) if the food grown each sol exceeds the food consumed.

Model Execution

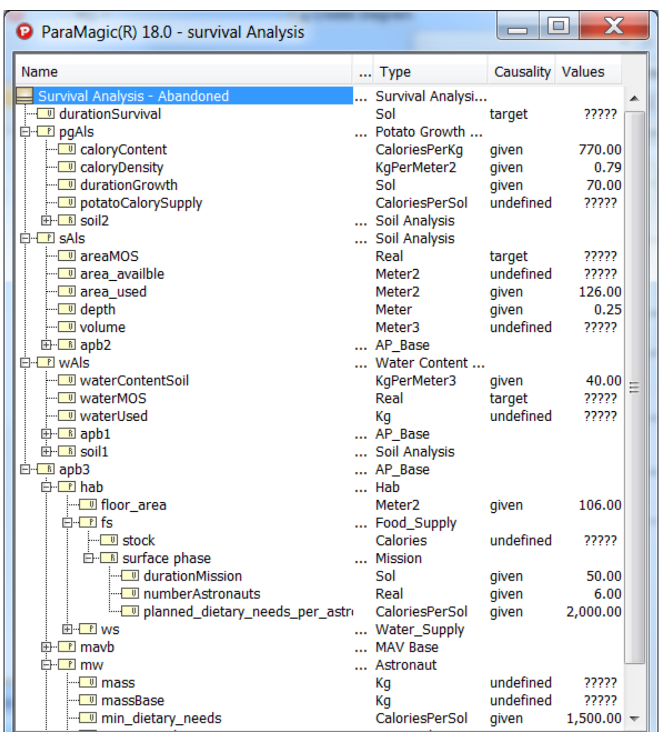

Figure 9 Instance01, before solving

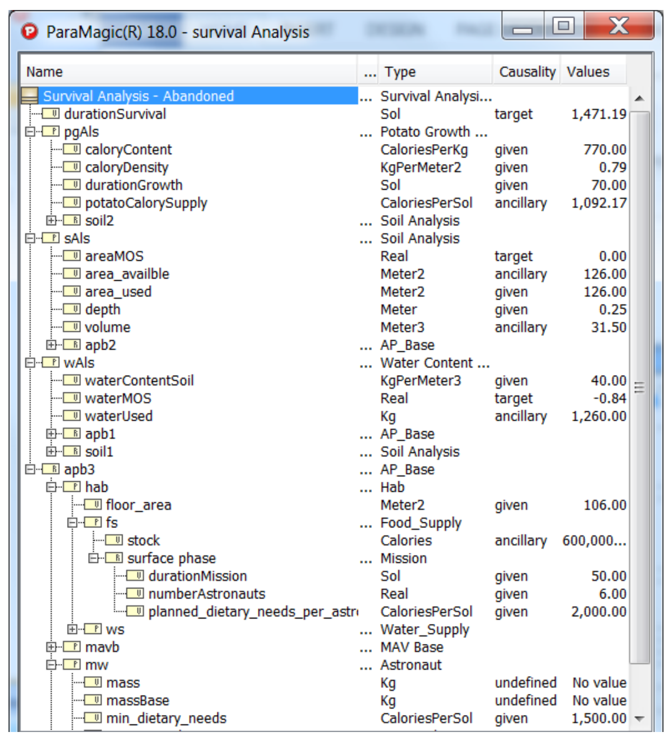

Figure 10 Instance01, after solving

Using values from (or estimated from) the Ares 3 mission, an instance of the parametric model is created and displayed in the ParaMagic browser in Figure 9 (before solution) and Figure 10 (after solution). Potatoes will extend the astronaut's survival from 400 sols to 1471 sols (if things go according to plan).

Martian Survival - Transit

The second scenario sends the astronaut on an extended trip across the Martian landscape. The top-level block in Figure 11, Caravan, consists of the two rovers coupled together, laden with everthing necessary to survive the trip. There is a time limit, by which the trip must be completed, and food and water stocks must be sufficient.

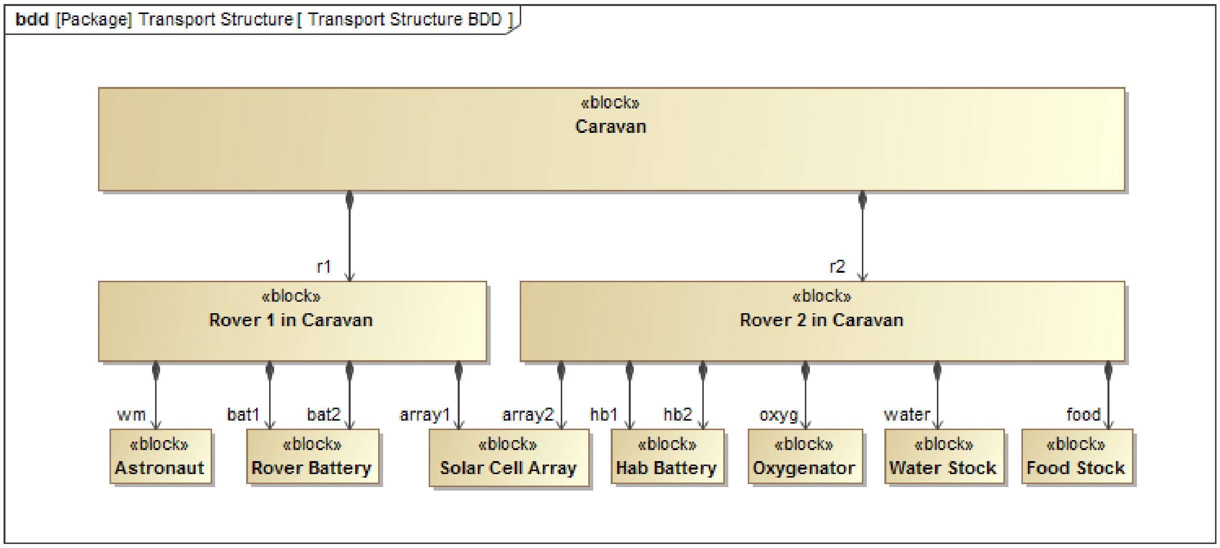

Figure 11 Transport Structure BDD, showing the caravan, rovers, and key cargo for the analysis

I use embedded parametrics in this part of the model for the mass roll up. All the blocks shown in Figure 11 are specializations of a supertype TravelElement, which uses recursion (Figure 12).

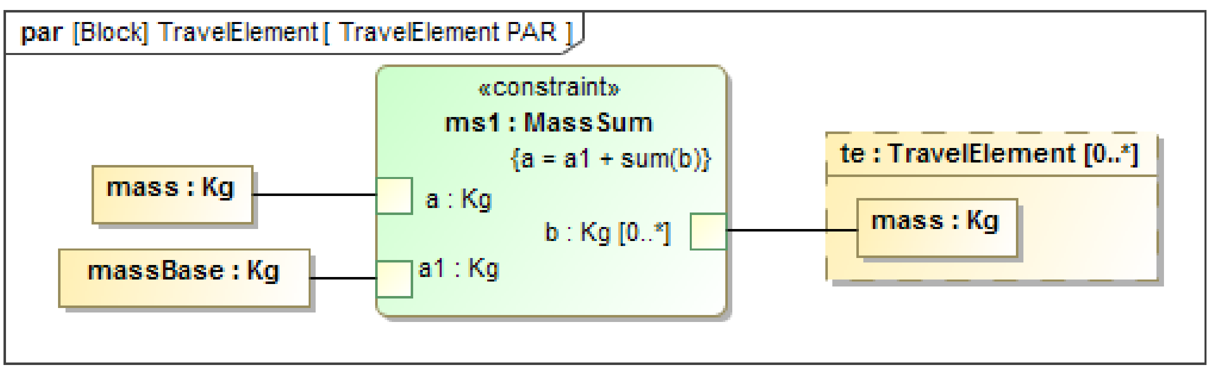

Figure 12 Parametric diagram for TravelElement. The mass of each element is the sum of the mass of its explicit parts plus a base mass

The astronaut drives each Martian day (sol) until the rover batteries are exhausted, manually unpacks and deploys the solar panels carried on each rover, waits while the panels recharge the batteries, and restows the panels, all during Martian daylight. Additional charging time is required to run the oxygenator periodically, which removes CO_2 from the rover air supply before it reaches toxic levels.

The number of solar panels to be carried on the trip represents a trade-off between the power produced, the mass to be transported, and the time spent unpacking and repacking the panels at each stop. These three values are captured in the parametric diagram in Figure 13.

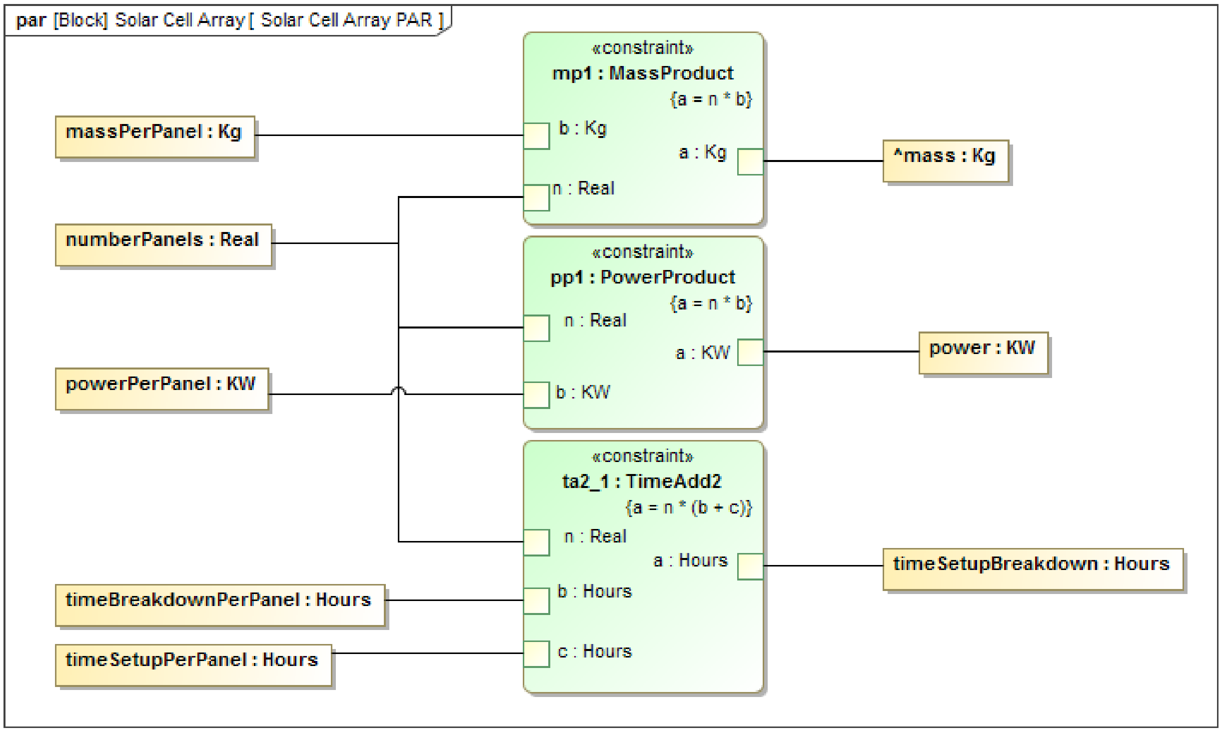

Figure 13 Solar Panel parametric analysis

As in the first scenario, we place most of the scenario specific analyses in separate blocks that reference the Caravan system (Figure 14).

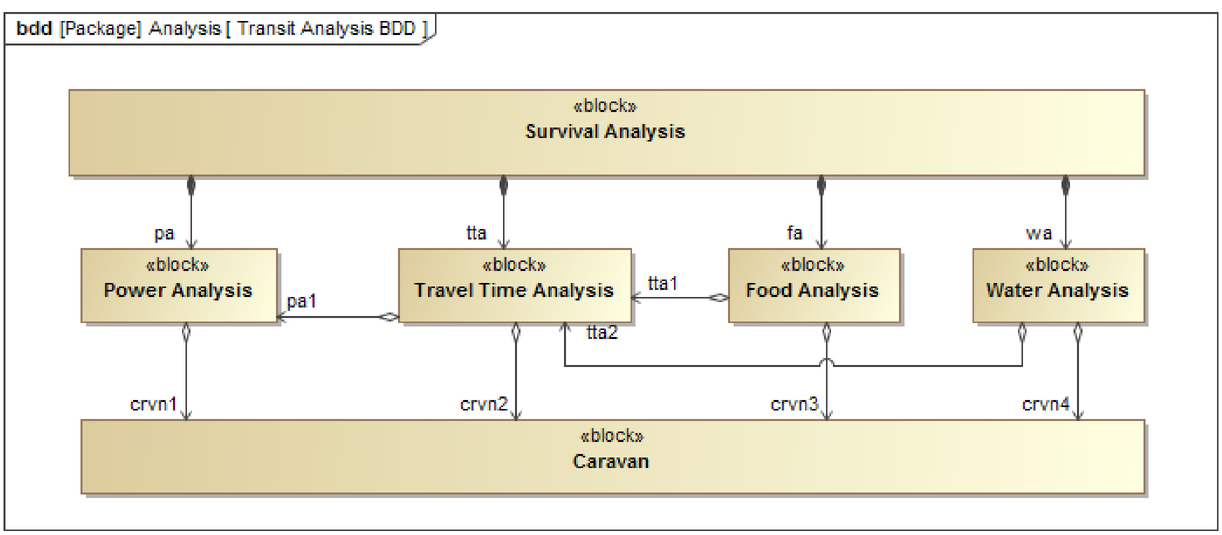

Figure 14 Transit Analysis BDD, showing the structure of the second scenario analyses

The Power Analysis PAR in Figure 15 calculates two key parameters, how long it takes to fully calculate the caravan batteries and what percentage of the time must be spent powering the oxygenator, relative to how long the atmosphere remains healthy. The second factor extends the total trip time by a fixed percentage, as shown in Figure 16.

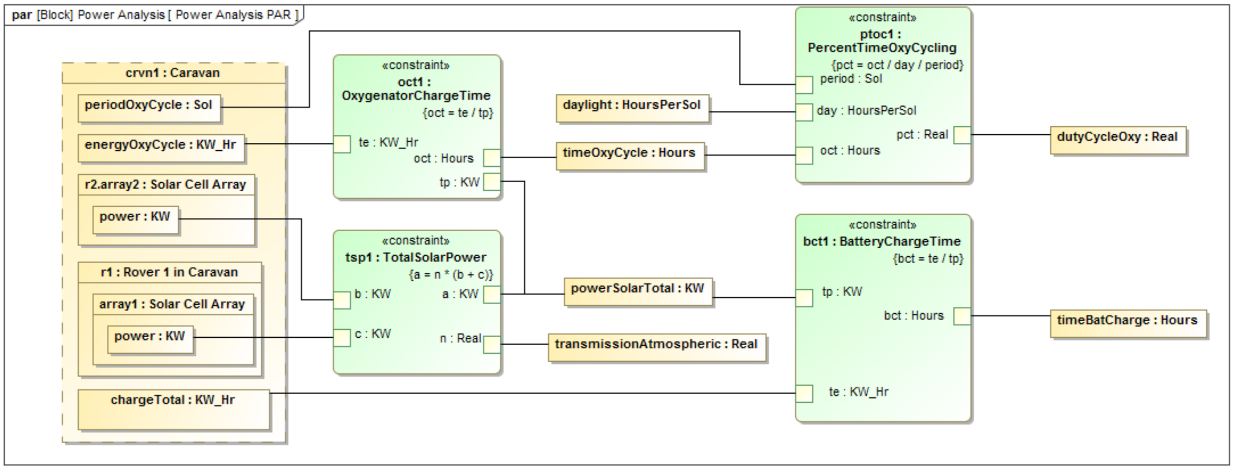

Figure 15 Parametric diagram for Power Analysis. Note that battery charging time is affected by atmospheric transmission factors, which may reduce solar panel output.

The travel Time Analysis PAR (Figure 16) uses the power numbers and the caravan properties to calculate total trip time. The constraint calculating distance traveled by the caravan per battery charge,

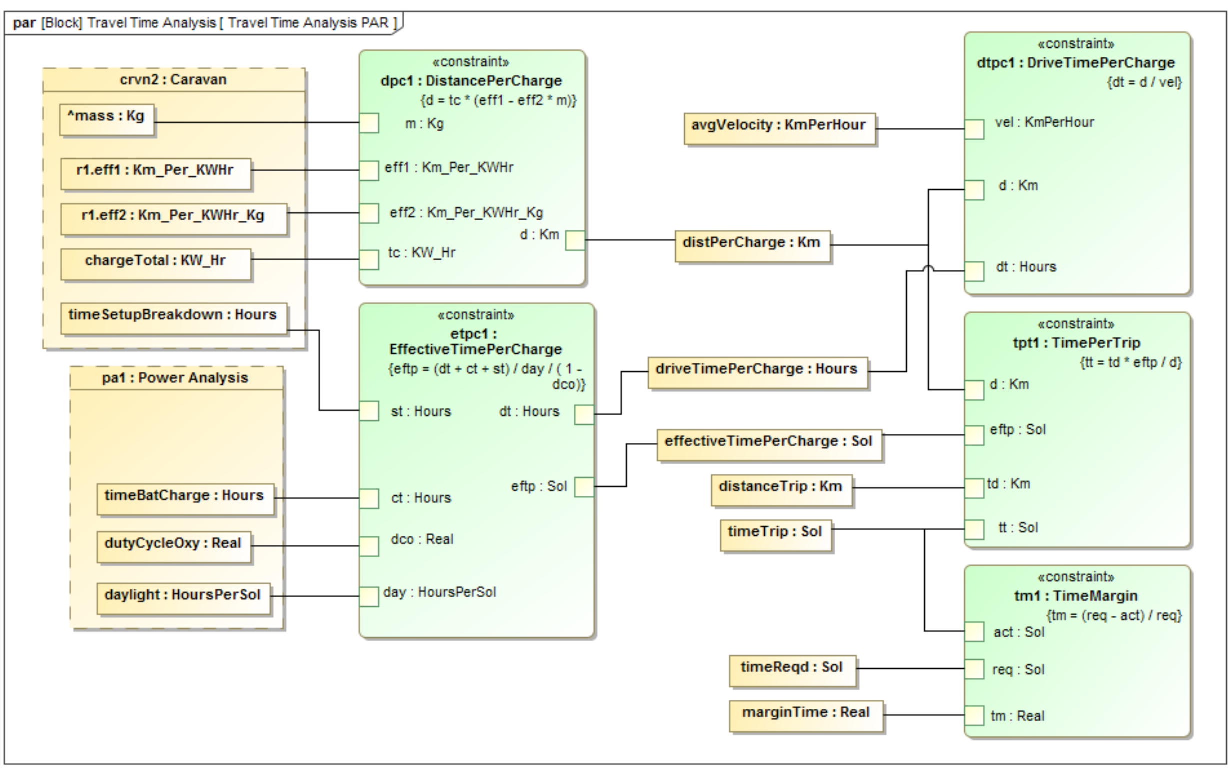

Figure 16 Parametric diagram for Travel Time Analysis. marginTime is the MOS for the calculated trip time vs. time limit

DistancePerCharge, is estimated from empirical data collected during the Ares 3 mission. This distance, divided by the effective time per charging cycle (the sum of the drive time, charge time, and solar panel unpack/repack time, extended by the daylight hours per sol and oxygenator operation), estimates an effective velocity including downtime. The total trip distance, divided by the effective velocity, results in timeTotal:sol, the total time for the trip.

Food and water calculations are similar to those in the first scenario and are not repeated here. At the Survival Analysis – Transit block (Figure 17) level, the parametric diagram uses conditionals to return verdicts for the three survival issues, 1 if the astronaut survives, 0 if he doesn't.

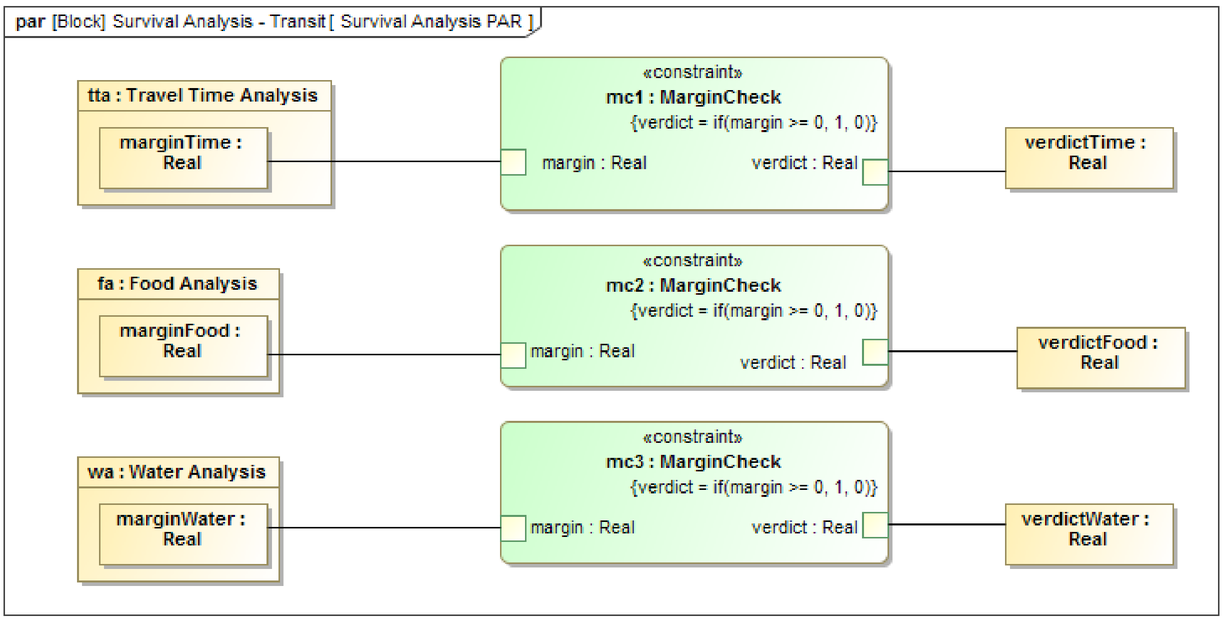

Figure 17 Parametric diagram for TravelElement. The mass of each element is the sum of the mass of its explicit parts plus a base mass

The instance values in Figure 18 and offer hope; all three factors are positive. The givens are taken from the Ares 3 video logs. The caravan can travel almost 100 kilometers pre charge, taking 4 hours at the maximum speed of 25 kph. Recharging the batteries takes about 10 hours of Martian daylight, with an additional 1.4 hours for setting out and picking up the solar panels. The caravan stops to power and run the oxygenator for about one sol out of every five. Overall, the trip of 3200 kilometers is expected take 47.55 sols, within the 50 sol time limit and with sufficient food and water. This analysis assumes no problems with terrain obstacles, equipment failure, weather, or any other challenge that Mars can throw at the unsuspecting visitor.

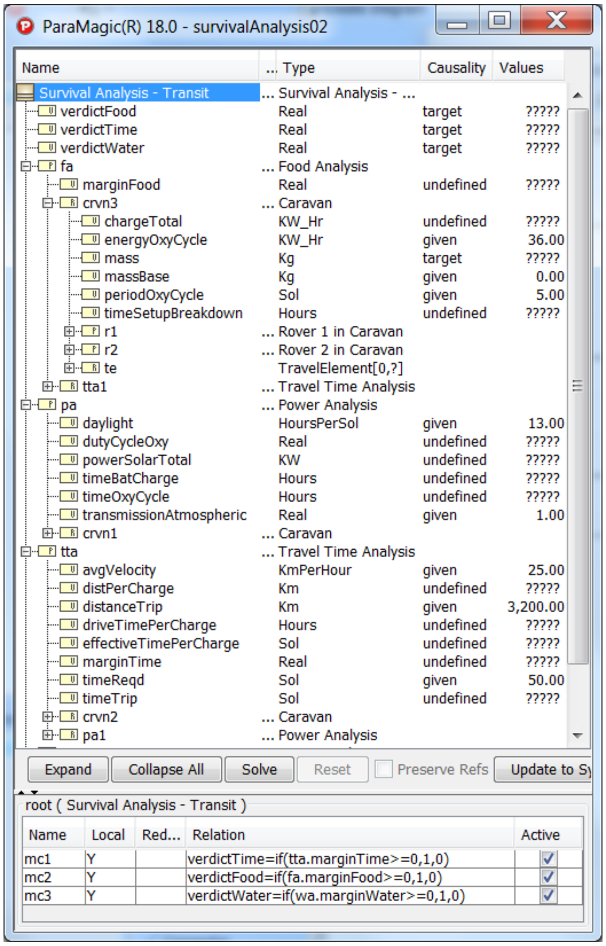

Figure 18 Instance02, before solving

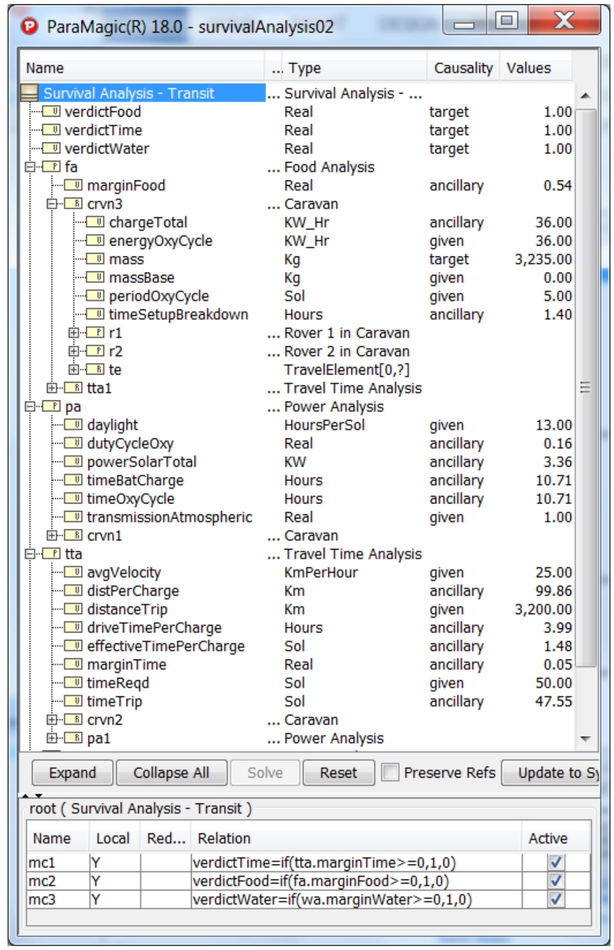

Figure 19 Instance02, after solving

Final Thoughts

A full treatment of the Ares 3 mission would start with modeling the original profile and treat the scenarios explored in this TechNote as extended use cases. Further extended use cases could be developed because (Spoiler Alert!) none of *The Martian*'s plans work out as expected. This further development would highlight the common strength of SysML and the astronaut hero, the ability to adapt and re-purpose. As the astronaut configures the elements of the original mission to serve his new objectives, the SysML modeler reuses object from old programs in new ways. I like to think that the NASA engineers working to rescue the hero were, behind the scenes, using MBSE to come up with the rapid redesigns they come up with.

About the Author

Dr. Dirk Zwemer (dirk.zwemer@intercax.com) is President of InterCAX LLC, Atlanta, GA and holds OCSMP certification as Model Builder - Advanced.

For further information, visit us at www.intercax.com or contact us at info@intercax.com.