# Behavior Pattern: Conceptual Examples v1.1

## Conceptual Examples

Two examples have been chosen to illustrate the use of the Behavior pattern:

- a flashlight example, chosen for its simplicity to introduce the concepts of the behavior pattern and demonstrate the flexibility of the pattern to modeling choices;
- a power/data model of a spacecraft, to demonstrate that the pattern can be used model complex behaviors relevant to JPL's field.

Table of Contents:



### Flashlight Example

Page Top | Next

The flashlight example consists of a simple model of the electrical circuit shown in the image on Figure 1. Three Components are identified:

- a battery;
- a switch;
- and a lamp.

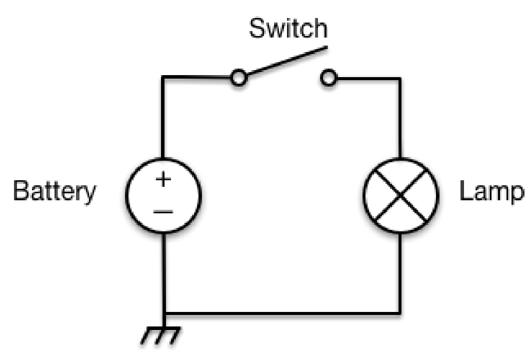


Figure 1. Flashlight electrical model

These three components are captured using the IMCE Mission ontology, more specifically as `m:Component`. Their respective functions are also captured using `m:Function`. For example, in the case of the battery, the identified function is to "provide electrical energy".

We introduce now concepts from the Behavior pattern. This example details the electrical aspect of the flashlight: as a consequence, the behavior is captured from the electrical point of view. In this first example, it was chosen to model this behavior using electrical state variables that represent voltage and current across the component. Another equivalent model is presented later in this page.

Page Top

### State Variables and Parameters

Page Top | Next

Two `StateVariables` of the lamp are shown in Figure 2, related to its electrical behavior:

- Voltage across the lamp (in effect an electrical potential difference between the two electrical terminals of the lamp)
- Current through the lamp

As defined in the ontology, `StateVariables` are related to their `Be` `havingElement` using the relationship `a:characterizes`.

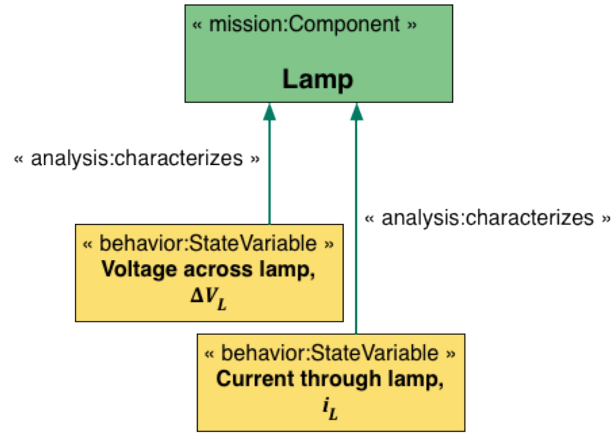


Figure 2. Example of `StateVariables` of the Lamp

Additionally, a `Parameter` is introduced to capture the Ohmic resistance of the lamp, as shown in Figure 3.

It is assumed in this example that the Ohmic resistance of the lamp remains constant, and therefore, a `Parameter` is the logical choice for capturing it. We could also have used a `StateVariable`, but this would have added unnecessary complexity. See Notes in the Conceptual Ontology page about usage choices between `StateVariables` and `Parameters`.

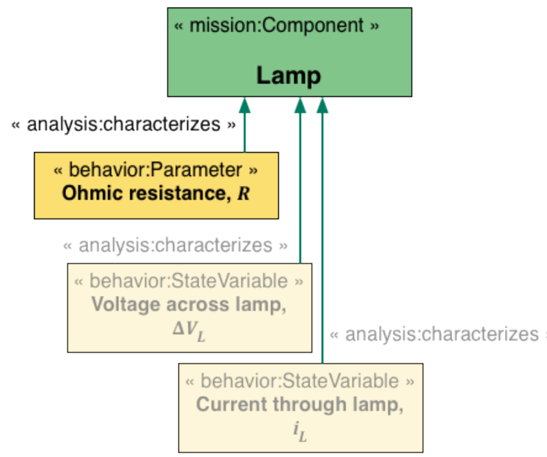


Figure 3. Example of `Parameter` of the Lamp

Page Top | Previous

## ElementBehavior

Page Top | Next

It is assumed that the lamp is of the incandescent type, and as such, it behaves as a Ohmic resistor. Based on the `StateVariables` and the `Parameter` of the lamp presented above, the associated `ElementBehavior` is modeled using Ohm's law. This is captured on Figure 4 by an equation that constrains the voltage and current `StateVariables`, and uses the resistance `BehavingElement`.

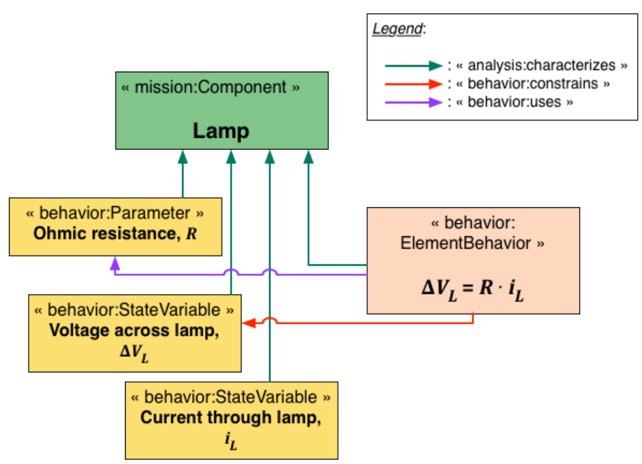


Figure 4. `ElementBehavior` according to Ohm's law

Figure 4 only presents a part of the lamp `ElementBehavior` we want to model. The lamp is an ohmic resistor with Joule heating that produces light due to the resulting thermal radiation. The Lumen output of the lamp is assumed to be equal to the electrical power of the lamp factored by the luminous efficacy of the glowing material (tungsten for example). The complete model of the lamp `ElementBehavior` is shown in Figure 5.

In this simple example, note that the lamp resistance and the luminous efficacy are assumed to be constant (they might not be in reality; for example, the resistance of an incandescent filament varies with temperature).

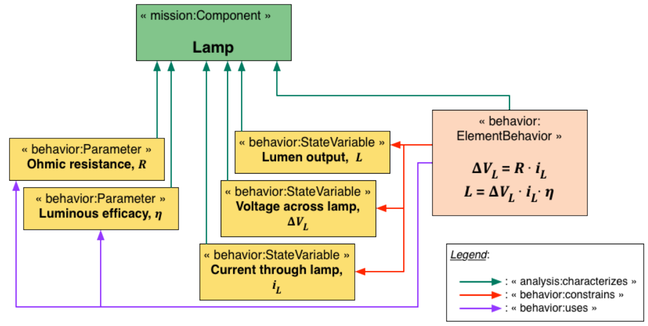


Figure 5. Complete lamp `ElementBehavior`

The same approach is taken for the other two components: the battery and the switch.

- Battery: a simple `ElementBehavior` model is retained for the battery: we assume that the voltage across the battery remains constant and at its nominal value. This is captured using a voltage `StateVariable` and a nominal voltage `Parameter`. The model is shown in Figure 5. Note that an extra `stateVariable` appears in Figure 6: this `StateVariable` is needed to model later the interaction between the three component.

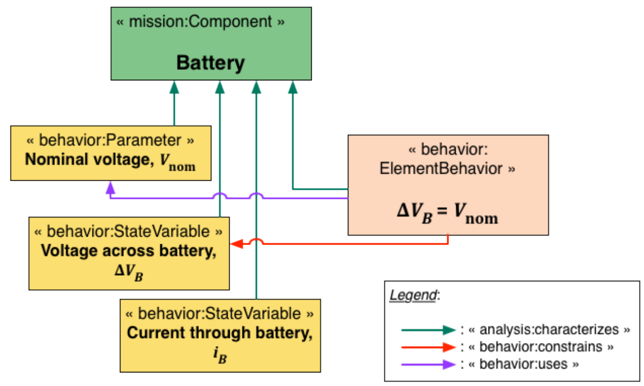


Figure 6. Battery `ElementBehavior`

- Switch: 3 `StateVariables` are introduced: Voltage and Current similarly to the battery, and a Switch position that reflects that the switch can be either open or closed. OPEN and CLOSED are two `States` of the Switch position state variable, and the `Codomain` of the switch position `StateVariable` is {OPEN,CLOSED}. We chose to capture the behavior of the switch as a state machine, shown in Figure 7. This state machine reflects the dynamics of the Switch position `StateVariable`:
    - if the switch is open, then the current through the switch is constrained to be zero;
    - if the switch is closed, then the voltage across the switch is constrained to be zero, assuming no electrical losses in the switch.
    - the transitions and the commands on the transitions are not explicitly part of the behavior pattern specification for now, but belong to the specification of the state machine semantics chosen (e.g., Harel statechart, SCXML, UML state machine). Here "cmd open" is the command/signal that triggers the transition from the CLOSED `State` to the OPEN `State` of the Switch position `StateVariable`, and the "cmd closed" is the command/signal that triggers the transition from the OPEN `State` to the CLOSED `State` of the Switch position `StateVariable`. The state machine semantics must be agreed upon between modelers, and it is not under the purview of the behavior pattern. In addition, future work on relating behavior pattern and inputs/commands is planned.

  Note: this state machine is an `ElementBehavior` as it represents the following constraint on `StateVariables`: "*if the switch position `StateVariable` is in the OPEN `State`, then the current `StateVariable` is zero; else if the switch position `StateVariable` is in the CLOSED `State`, then the voltage `StateVariable` is zero*".

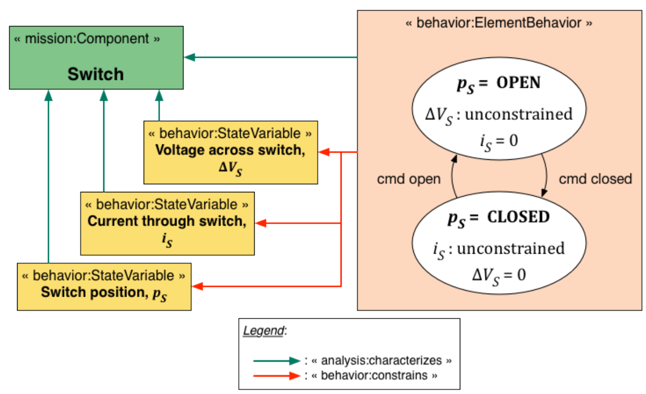


Figure 7. Switch `ElementBehavior`

Page Top | Previous

## Interaction

Page Top | Next

The last step in capturing the complete behavior of the flashlight is to model the electrical interaction between the three components. Given the choice of the state variables to characterize the components, the interaction between the component is captured on a logical level by considering the electrical loop formed by these components (mesh analysis approach). The following details this approach.

### Interaction

As mentioned above, the interaction between the component is captured on a logical level by considering the electrical loop formed by these components (mesh analysis approach): the three components interact through the `Interaction` representing the electrical loop. Figure 8 shows the `Interaction` joining the three `BehavingElements`.

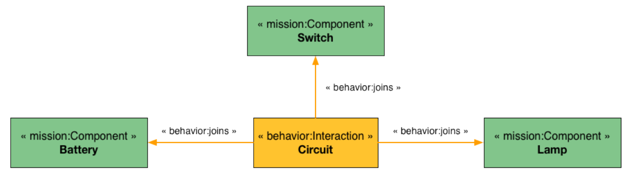


Figure 8. Electrical circuit `Interaction`

### InteractionBehavior

In the electrical loop, the current must be the same at every point, and Kirchhoff's voltage law applies ("the directed sum of the electrical potential differences across any closed network is zero", Kirchhoff's circuit laws, *Wikipedia*). These two constraints on the voltage and current `StateVariables` are captured in the `InteractionBehavior` shown in Figure 9.

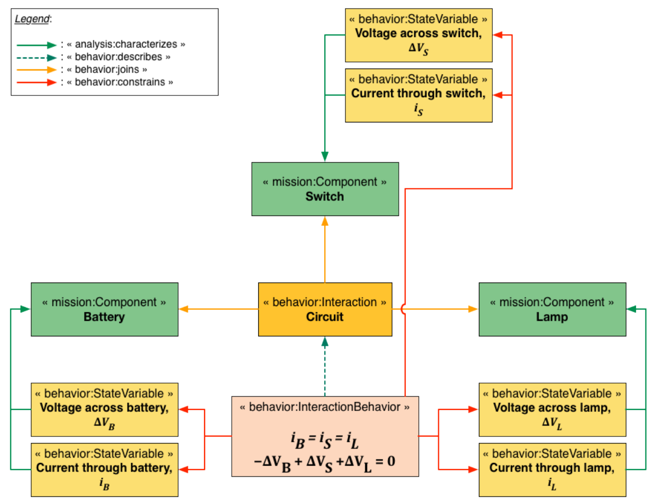


Figure 9. Mesh analysis `InteractionBehavior`

Page Top | Previous

## Complete Flashlight Model for a Mesh Analysis

Page Top | Next

The complete model of the flashlight combining Figures 1 to 9 is shown in Figure 10. Note that the grey boxes in Figure 10 do not represent an ontological concept; their only purpose is for readability and to highlight graphically concepts related to each component.

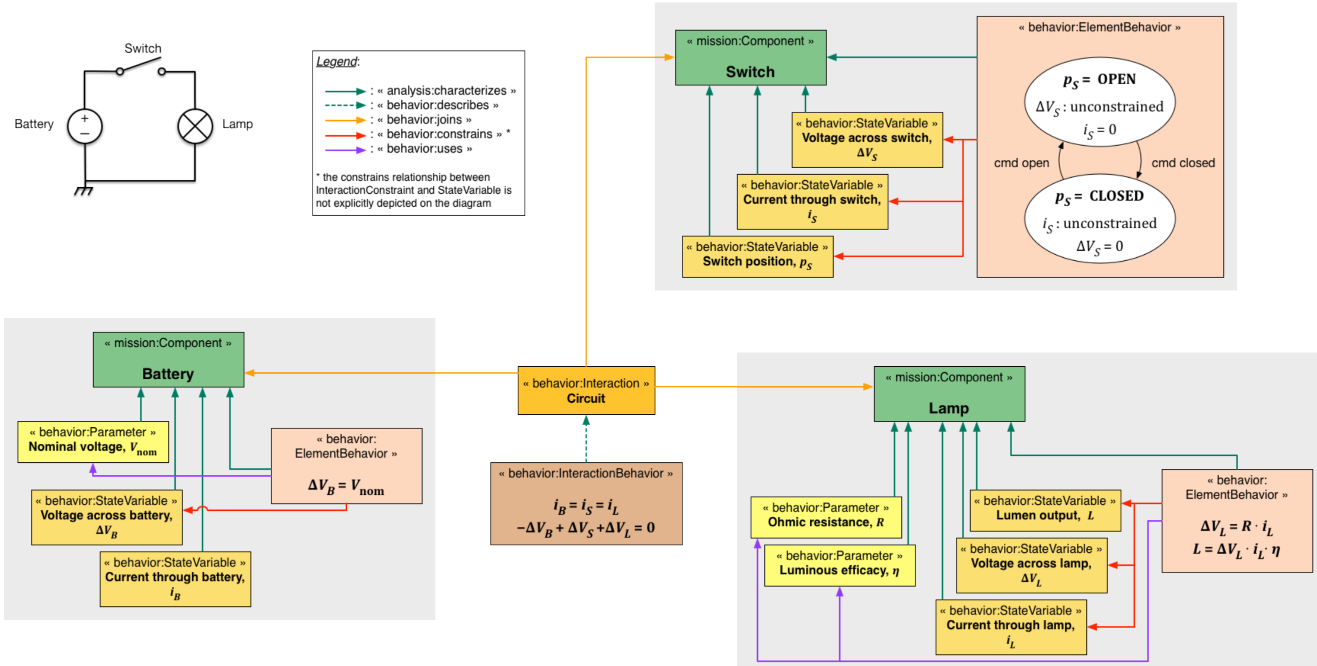


Figure 10. Flashlight behavior model

Page Top | Previous

## Scenario and Trajectory

Page Top | Next

This model captures sufficient information to be able to simulate `Trajectories` of the `StateVariables` given a `Scenario`. A simple example `Scenario` is considered here: the switch is in the open position until 7pm (flashlight does not emit light), where the switch is commanded closed for 5 minutes, then switched back to the open position. In effect, the user needed light from the flashlight between 7 and 7:05pm. Assuming some values for the three `Parameters` of the model, the `Trajectories` of the seven `StateVariables` can be simulated by solving the constraint equations of the `ElementBehavior` and `InteractionBehavior` blocks. The `Scenario` and the `Trajectories` are shown in Figure 14. The `Scenario` is described using a State Analysis representation, where the grey circles represent time points and the green rectangles represent goals. The reader is referred for more details to the paper by Ingham et al, "Engineering Complex Embedded Systems with State Analysis and the Mission Data System," *Journal of Aerospace Computing, Information, and Communication*, 2, 2005.

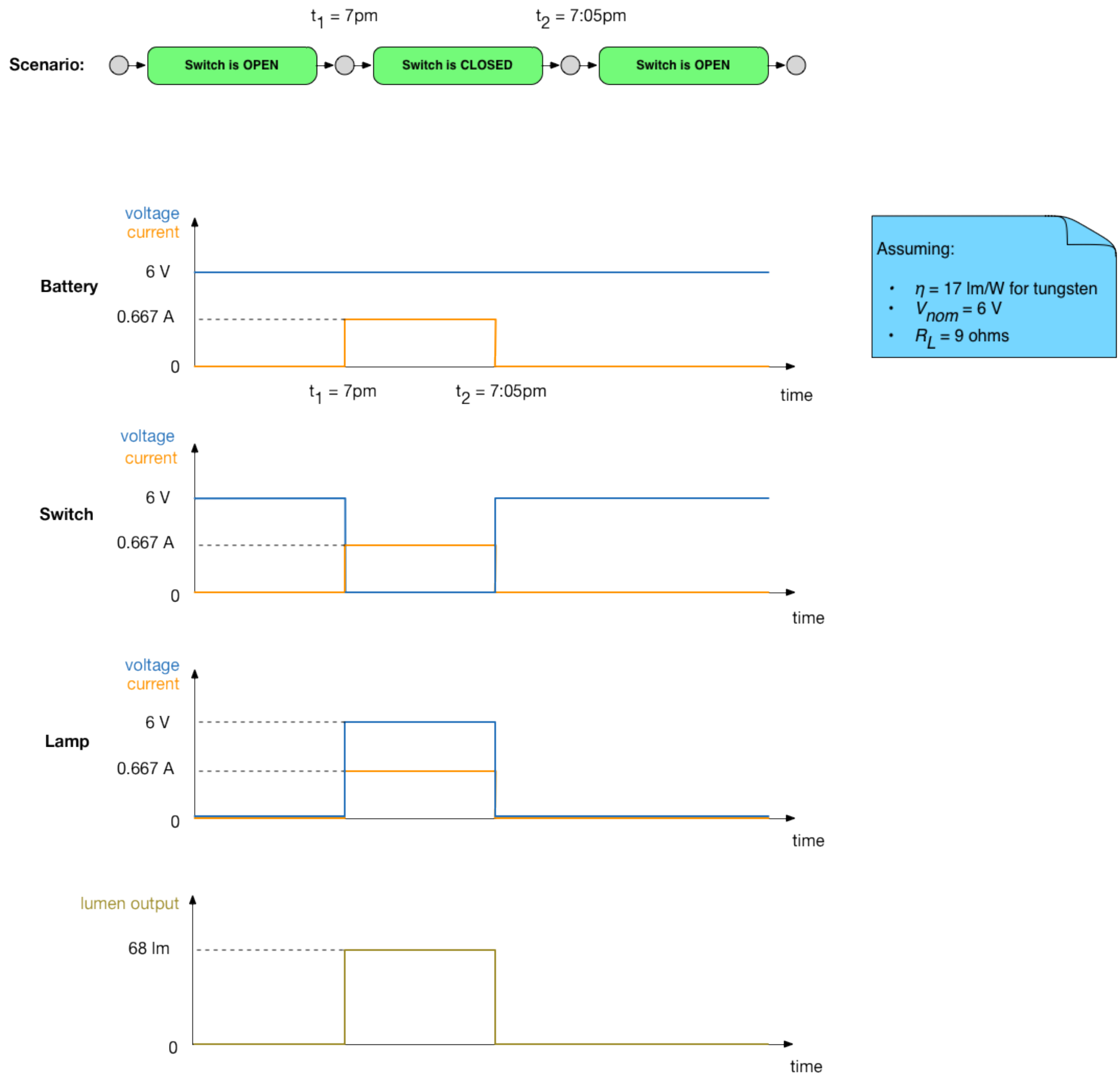


Figure 11. `Scenario` and `Trajectories` for the flashlight example

Page Top | Previous

## Alternative Behavior Model for the Flashlight

Page Top | Next

In the previous section, a simple behavior model was introduced to give examples for the concepts defined in the behavior pattern. The model was also chosen to mimic how one could go in solving for the current and voltage of the components in this electrical circuit using a mesh analysis approach. However, this approach does not reflect the actual physical electrical connection of the circuit. The interaction was captured from a logical point of view. In this section, it is shown how to apply the behavior pattern to capture a physical point of view that also shows the flexibility of the behavior pattern with respect to modeling choices. This alternative approach would also reflect how one would solve for the current and voltage of the components using a nodal analysis approach, instead of a mesh analysis approach.

This alternative model is shown in Figure 12 and is explained below.

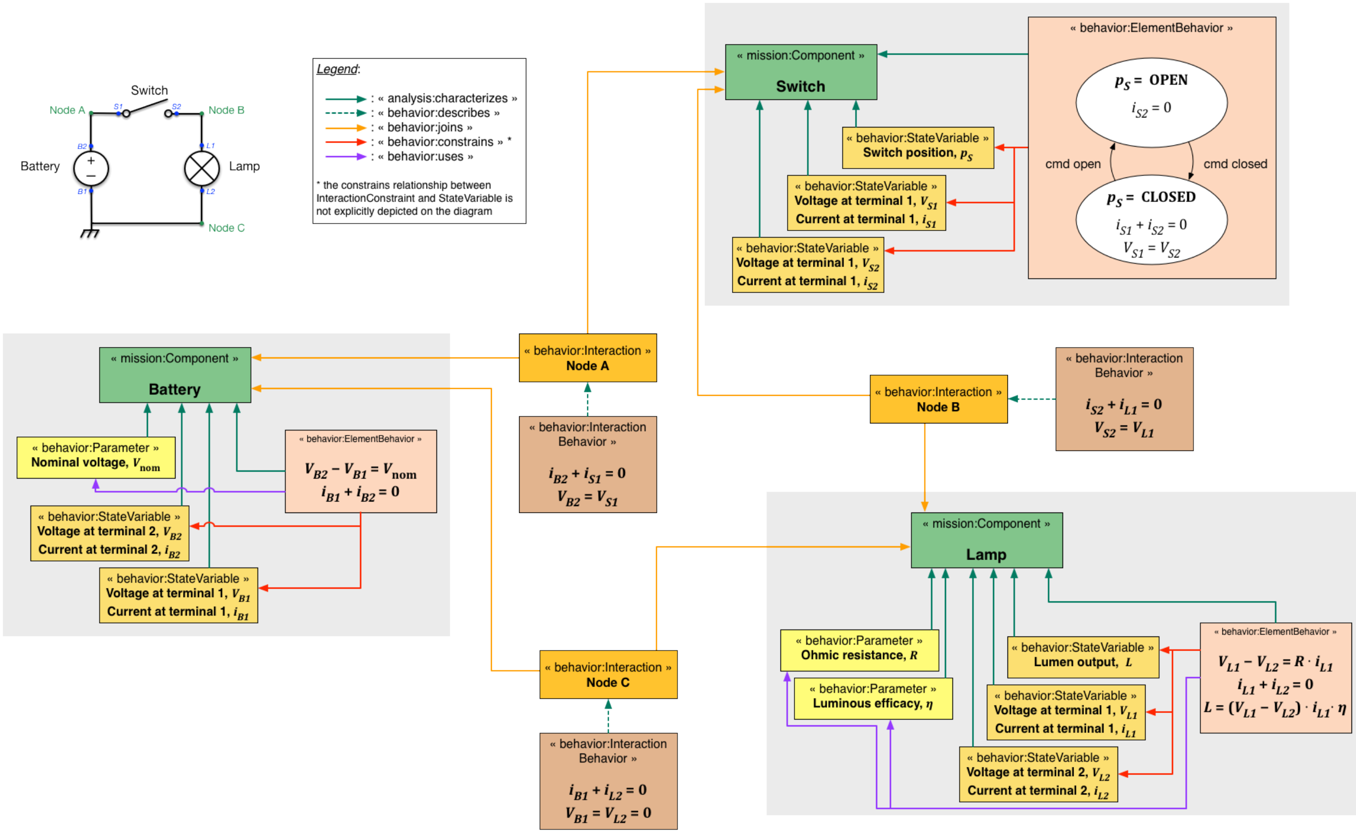


Figure 12. Alternative behavior model for the flashlight

In this model, the voltage and current of each component is taken at their two electrical terminals (these are not `InteractionTerminals`, but the physical electrical terminals). For example for the battery, $V_{B1}$ is the voltage at the battery terminal B1 connected to the terminal L2 of the lamp through Node C (also connected to the ground), and $V_{B2}$ is the voltage at the battery terminal B2 connected to the terminal S1 of the switch through Node A. The same is valid for the currents: $i_{B1}$ is the incoming current at the battery terminal B1 and $i_{B2}$ is the outgoing current at the battery terminal B2. Incoming currents are counted positive, outgoing currents negative. Notice that the number of `StateVariables` close to doubled in this example compared to the previous model.

The `StateVariables` are different in this model, so the constraint equations for the `ElementBehavior` of each component are also different. For example, in the case of the battery, two constraints are now necessary: one for the currents and one for the voltages. For the current, the equation $i_{B1} + i_{B1} = 0$ reflects the conservation of the electrical flux of electrons, and $V_{B2} – V_{B1} = V_{\text{nom}}$ states that the difference in electrical potential between the two battery electrical terminals is equal to the nominal voltage of the battery (same simple model than in the previous example). The behavior equations of the switch and the lamp are similarly adapted.

Due to a more physical approach taken in this mode, three `Interactions` must be considered: the one between the battery and the switch, the one between the switch and the lamp, and the one between the lamp and the battery (and the ground). The `InteractionBehavior` equations attached to the `Interaction` blocks reflects the electrical connections between the electrical terminals: the electrical charge is conserved according to Kirchhoff's current law ("algebraic sum of currents in a network of conductors meeting at a point is zero", Kirchhoff's circuit laws, *Wiki pedia*), and the electrical potentials (voltages) are the same.

Of course, equivalent `Trajectories` can be found for this set of state variables compared to the previous model given the same `Scenario`.

Page Top | Previous

## Alternative Model with InteractionTerminal

Page Top | Next

As discussed in the conceptual ontology page, `InteractionTerminals` are proposed to the model as optional. The examples above show the recommended practice, and this section describes how to use `InteractionTerminals`, should the modeler wishes to experiment with them. InteractionTerminals expose the StateVariables or Parameters that are involved in the interaction. In the case of the flashlight example, the voltage and current StateVariables of each component are exposed through InteractionTerminals, as these two types of StateVariables will be constrained in the interaction. The lumen output of the lamp for example is not involved in this interaction, and thus is not exposed by the InteractionTerminal of the lamp.

For example, the battery presents its InteractionTerminal that exposes the battery voltage and current StateVariables as shown in Figure 8.

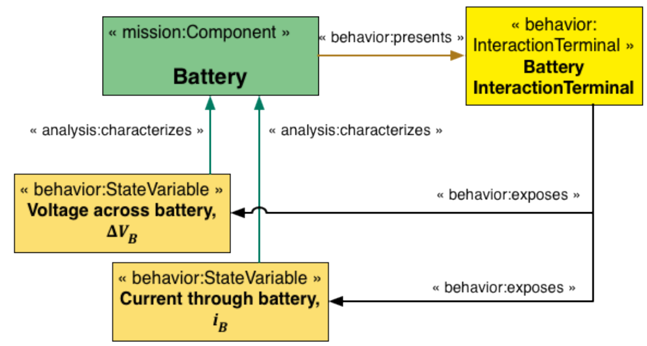


Figure 13. Battery InteractionTerminal

Figure 14 shows the complete alternate model (as in Figure 12) using InteractionTerminals. Each component presents two InteractionTerminals, one for each of its electrical terminals and each component's InteractionTerminal is joined by a different Interaction block than the other. For example for the Battery InteractionTerminals, the "Battery Terminal 1" one is joined by the Interaction "Node C", while the "Battery Terminal 2" one is joined by the Interaction "Node A". Each of these InteractionTerminals exposes the appropriate current and voltage state variables of the associated terminal.

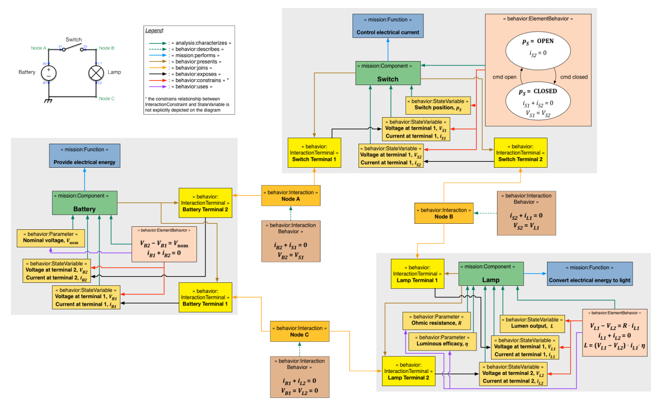

Figure 14. Model using InteractionTerminal

Page Top | Previous

## Spacecraft Power and Data Model

Page Top

A second example was selected to demonstrate the capability of the behavior pattern to handle more complex component behaviors and interactions. This example presents the behavior model of two spacecraft instruments ( a camera and a magnetometer) and two related spacecraft subsystems: the power subsystem and the data handling subsystem. The behavior model of this example is presented in Figure 16, and explained below.

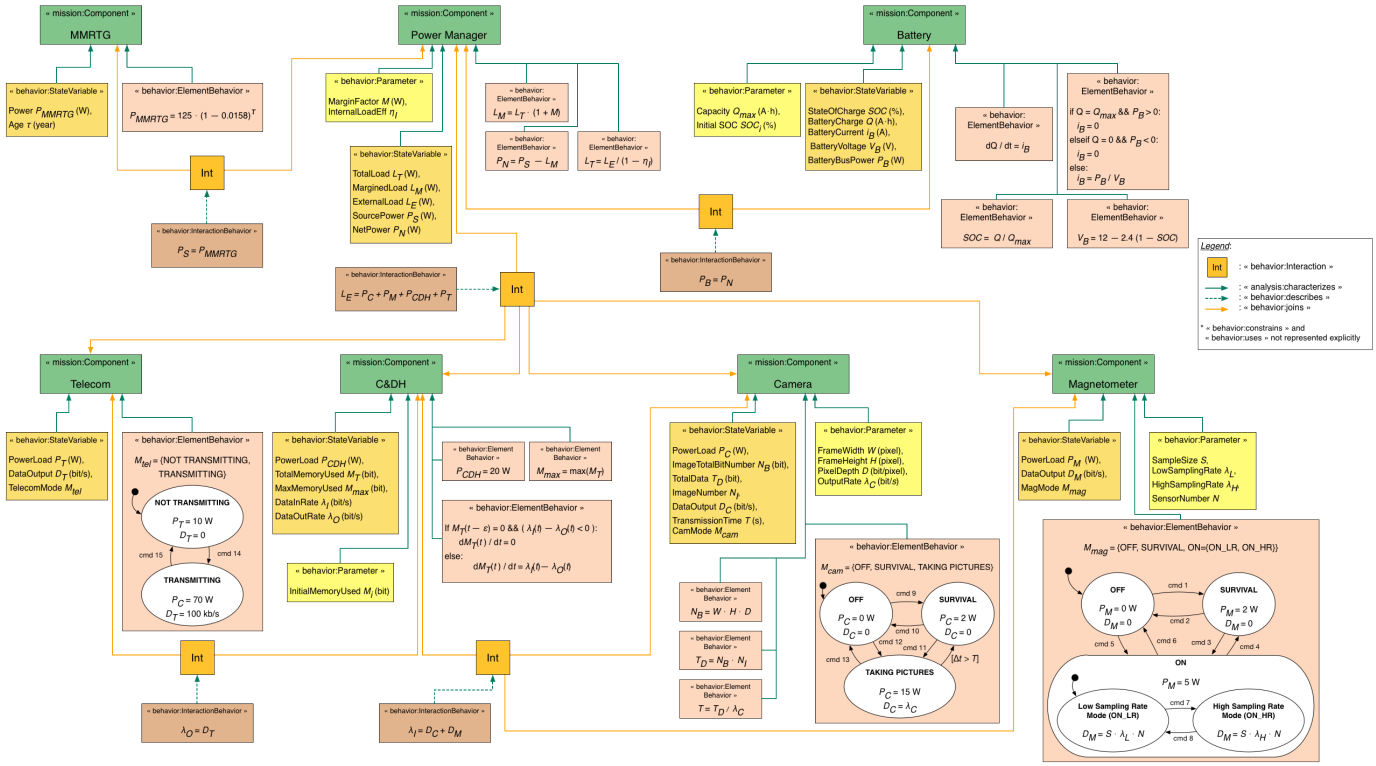

Figure 16. Behavior model of the spacecraft example

In this model, 7 `m:Components` are considered, and shown in green in Figure 16:

- the magnetometer instrument
- the camera instrument
- a primary power source using a MMRTG
- a battery used when power demand exceeds the capability of the MMRTG
- the power system that acts as the medium between the MMRTG power source, the battery, and the power loads (instruments and data subsystem)
- the C&DH that manages the storage of the data generated by the two instruments
- the Telecom that transmits the data stored by the C&DH

In this example, the behavior of some components captures more complex dynamics than in the flashlight example (such as for the battery model). The `ElementBehaviors` are briefly described below:

- Magnetometer: the magnetometer behavior captures four states of the magnetometer: OFF, SURVIVAL or ON at a low sampling rate, and ON at a high sampling rate (the survival state indicates that the survival heaters of the magnetometer are turned on). For each state, an estimated power load has been captured (i.e., the magnetometer consumes 0 W when turned off, but 5 W when turned on). A similar approach for data generation has been taken: for each state, an associated generated data rate has been determined. For example, when on, the magnetometer data output rate is a function of the number of sensors, the sampling rate associated with the current sampling rate mode (low or high) and the sampling size. The transitions between states have been captured as commands, but have not been elaborated further.
- Camera: the camera behavior captures three states of the camera: OFF, SURVIVAL or TAKING PICTURES (the survival state indicates that the survival heaters of the camera are turned on). For each state, an estimated power load has been captured (i.e., the camera consumes 0 W when turned off, but 15 W when actively taking pictures). A similar approach was taken for the data generation capture: for each state, a data output rate has been determined. In particular, a choice was made to model the behavior of the camera using a data rate based on a data volume determined by the number of pictures taken. This is particularly of interest for transitioning out of the Taking Pictures state: this transition is based on the time necessary to transmit the generated data volume based on a give data rate output. Other models for the camera can be envisioned.

- Let's now look into the power aspect of this architecture:
  - the C&DH is assumed to be always on and consuming 20 W of power. More detailed power load behavior could be supported by the behavior pattern.
  - The Telecom has two power states: transmitting or not transmitting with associated power loads.
  - The MMRTG power behavior is modeled using an exponentially decaying model. It produces 125 W of power at the beginning of life and 100 W after 14 years. This model is based on the AIAA paper by A. K. Misra, entitled "Overview of NASA Program on Development of Radioisotope Power Systems with High Specific Power," presented at the 4th International Energy Conversion Engineering Conference and Exhibit (IECEC), San Diego, CA, 2006.

- The power system behavior captures the power margin and the internal losses on top of the total power load of the spacecraft. The net power is then captured by subtracting the total load to the available power.
- The battery behavior model is more complex than the one presented in the flashlight, as it takes into consideration the state of charge of the battery, and its influence on the voltage and current of the battery. The battery can be charging or discharging based on the battery bus power sign.
- The information above captured the `ElementBehavior` of each of the components. The following describes how these components interact with one another from a power perspective. Three interactions are modeled:
  - The interaction between the MMRTG and the power system: this interaction indicated that the power produced by the MMRTG is equal to the available source power in the power system.
  - The interaction between the power system and the four power loads (Telecom, C&DH, camera and magnetometer). It indicates that the total power load is the sum of each of the power load of the four components.
  - The interaction between the power system and the battery. It states that the net power (available power minus total load) is equal to the power that the battery experiences: if the net power is negative (the MMRTG is not providing enough power for the load current demand), then the battery provides that extra power (within the limits of its capacity) and is thus discharging; if the net power is positive, then the battery uses this extra power to recharge if necessary.

- Finally, let's have a look at the data aspect of this architecture:
  - The C&DH commits to memory the data its receives and remove from memory the data it outputs for transmission. This model does not capture the practice of keeping data in storage until the ground had acknowledge reception.
  - As described above, the Telecom has two states: a non-transmitting state, and a transmitting one with a fixed data output rate.
  - The information above captured the `ElementBehavior` of each of the components. The following describes how these components interact with one another from a data perspective. Two interactions are modeled:
    - The interaction between the C&DH and the two instruments: it states that the data the C&DH receives is the sum of the data generated by the two instruments.
    - The interaction between the C&DH and the Telecom: it states that the data the Telecom transmits is the data the C&DH outputs.

This model captures sufficient information to be able to simulate trajectories of state variables given a scenario. An example scenario is considered here: the camera and the magnetometer are turned on and back to survival mode according to a specified schedule, and the Telecom transmitting all of the data generated. Assuming some values for all `Parameters` of the model, the trajectories of several `StateVariables` can be simulated by solving the constraint equations of the `ElementBehavior` and `InteractionBehavior` blocks. The `StateVariables` of interest in the `Scenario` considered in this example are the power of the MMRTG, the total load with margin, the individual power loads, the battery state of charge, and the total memory used by the C&DH. The `Scenario` is shown in Figure 17 and the resulting `Trajectori`ers are shown in Figure 18. The `Scenario` is captured using again a State Analysis representation, the reader is referred for more details to the paper by Ingham et al.

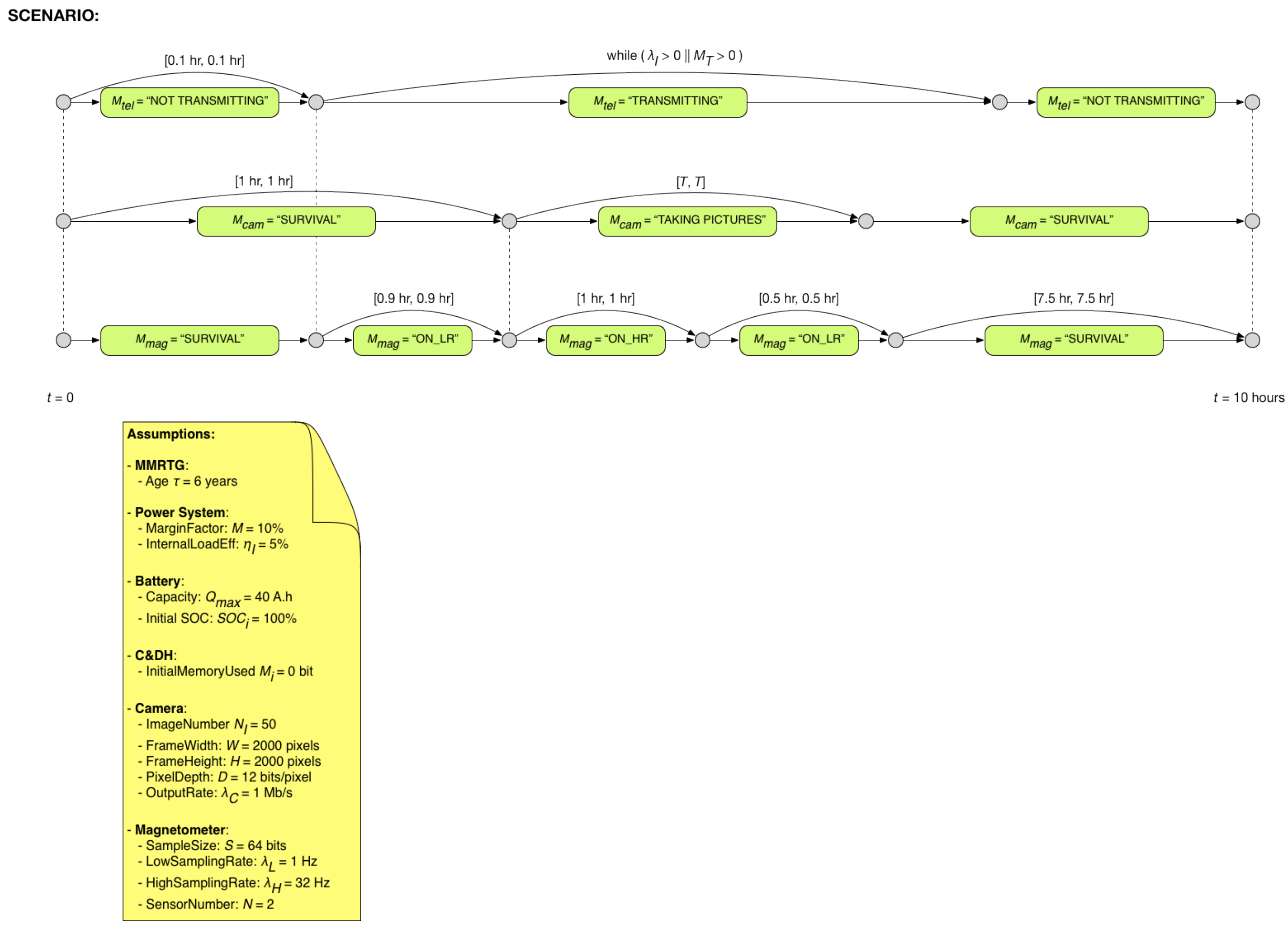


Figure 17. Example `Scenario` for the spacecraft model

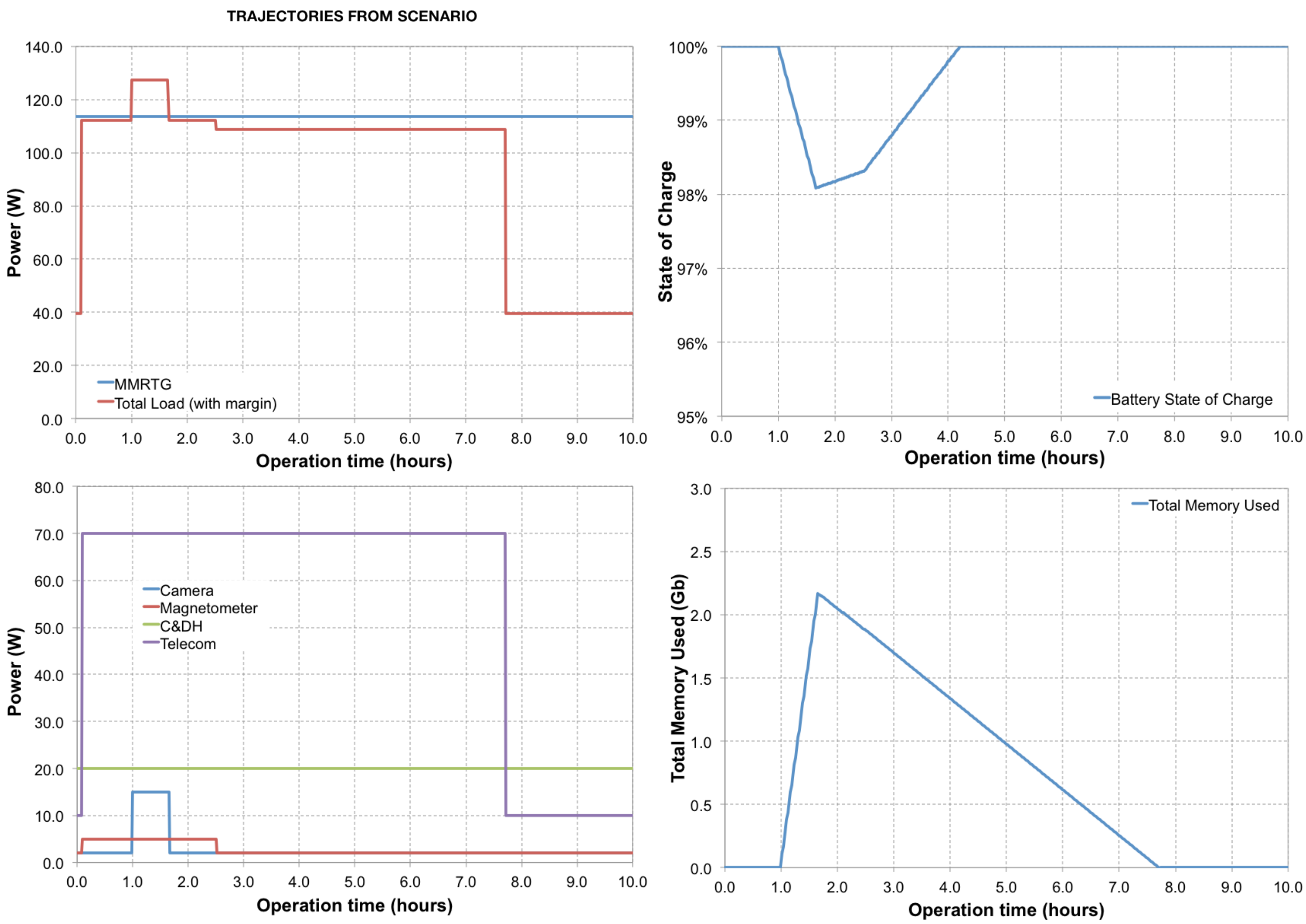


Figure 18. `Trajectories` of `StateVariables` of interest based on the example `Scenario`

Page Top | Previous

***Behavior Page Navigation - continue reading:***

(0) Community Page

(1) Main Behavior Pattern Page

(2) Conceptual Behavior Ontology v1.1 (3) Behavior Pattern: Conceptual Examples v1.1

(4) SysML-Embeddable Ontology & Implementation v1.1 (5) Behavior Pattern: SysML Example v1.1