The SysML Parametric Diagram

Lecture 111, v02

Index

Background

Background

- Constraints are limitations placed on the design of a system
- If designers had no constraints on the design to deal with, we could achieve anything, even that which extends beyond the constraints of physics
  - Not to mention not having to deal with business constraints such as getting the job done within a certain among of time and within a certain budget
- System designers have very large numbers of constraints to live within, and many of those are defined in the system requirements
- Other constraints are those that apply universally to everyone, such as constraints we live within in the physical world (gravity, for example)
- Other constraints are imposed upon us by government or societal policies and customs
- The kind of constraints we are interested in within Systems Engineering are those that can be expressed as equations (relationships) of some kind and, when analyzed, reveal the performance potential of the system, as designed

Purpose

Purpose

- The purpose of the parametric diagram is to
  - Describe the binding of constraint parameters to system value properties
    - The system value properties provide input parameter values that are used in parametric constraints (equations)
    - The results of the parametric analysis can also be provided to system or analytical value properties to report the results of the analysis
  - Describe the binding of one constraint parameter related to one constraint (equation) to another constraint parameter in another constraint (equation)
- These constraints are represented in SysML as “constraint blocks”
  - As with all elements in the system, these block need to be defined in a Block Definition Diagram (BDD)
- The system and analytical value properties that feed the parametric analysis or that receive results from the analysis must also be defined on a BDD
- So, there is quite a bit of SysML model development that needs to be performed before the definition of the parametric analysis is fully formed

Performing Parametric Analysis

- The progression of capability maturity in systems modeling typically develops as follows (developed by JG Artus):
  - Level 0 - Does not use modern system modeling tools to define the system
  - Level 1 - Uses system modeling tools to define the component and functional structure of the system
  - Level 2 – Uses system modeling tools to define the system behavior
  - Level 3 - Uses system modeling tools to perform allocation of functions to components and to show satisfaction of requirements by system elements
  - Level 4 - Uses system modeling tools to define system parametrics, MOEs, MOPs, etc
  - Level 5 – Uses system modeling tools along with parametric solvers* to execute parametric analysis of the system
- As you can see, the holy grail of Systems Engineering Modeling is to reach the point of performing a parametric analysis of the system in order to identify the best performing parametric configuration of the system
- This is difficult to achieve and requires a deep modeling infrastructure in the organization and a good deal of data to support such an analysis

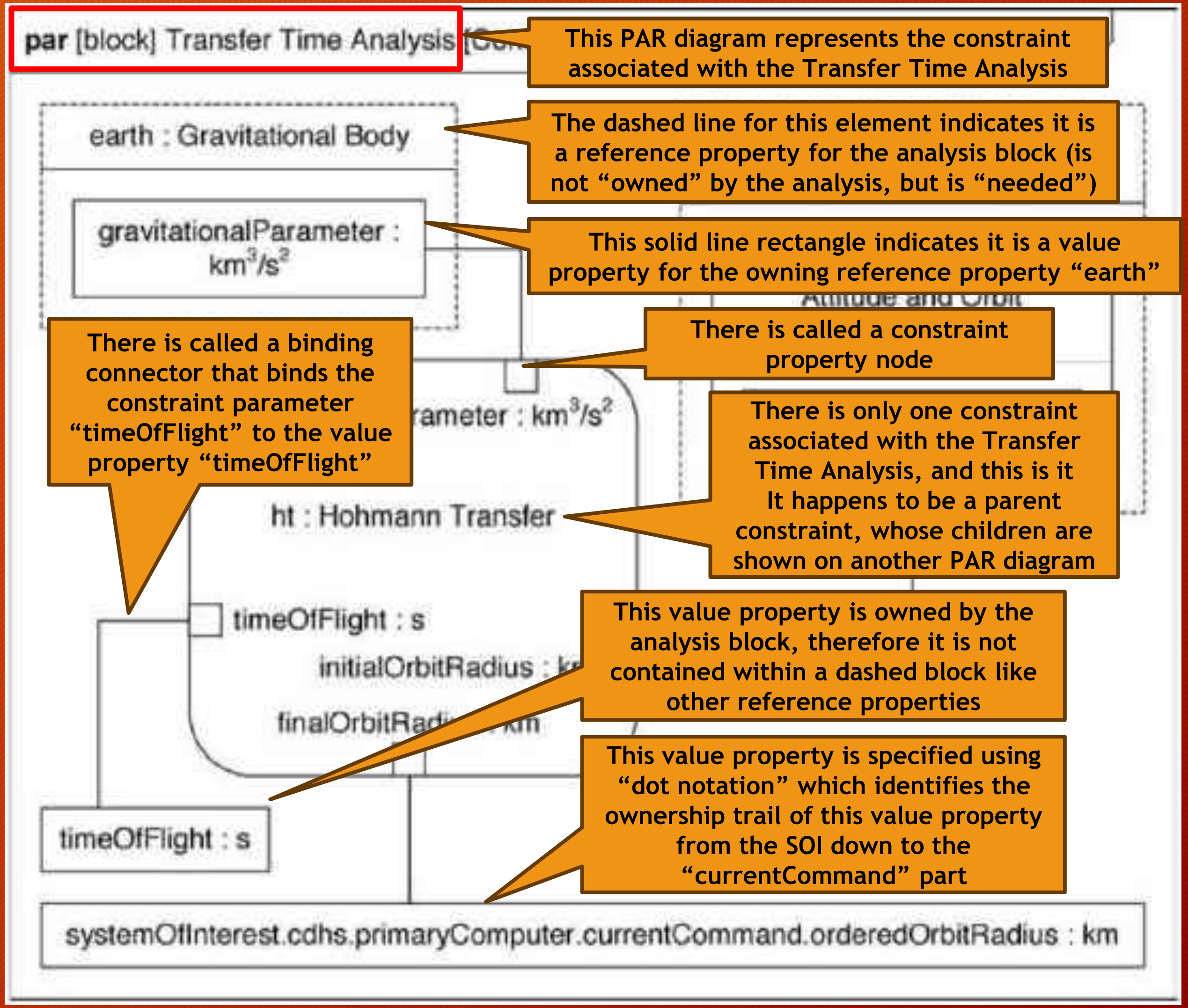
* Most system modeling tools require purchase of a separate parametric solver to perform this type of analysis

The capability maturity model of systems modeling shown here was developed by JG Artus in 2024

Parametric Diagram Syntax and Semantics

Syntax and Semantics

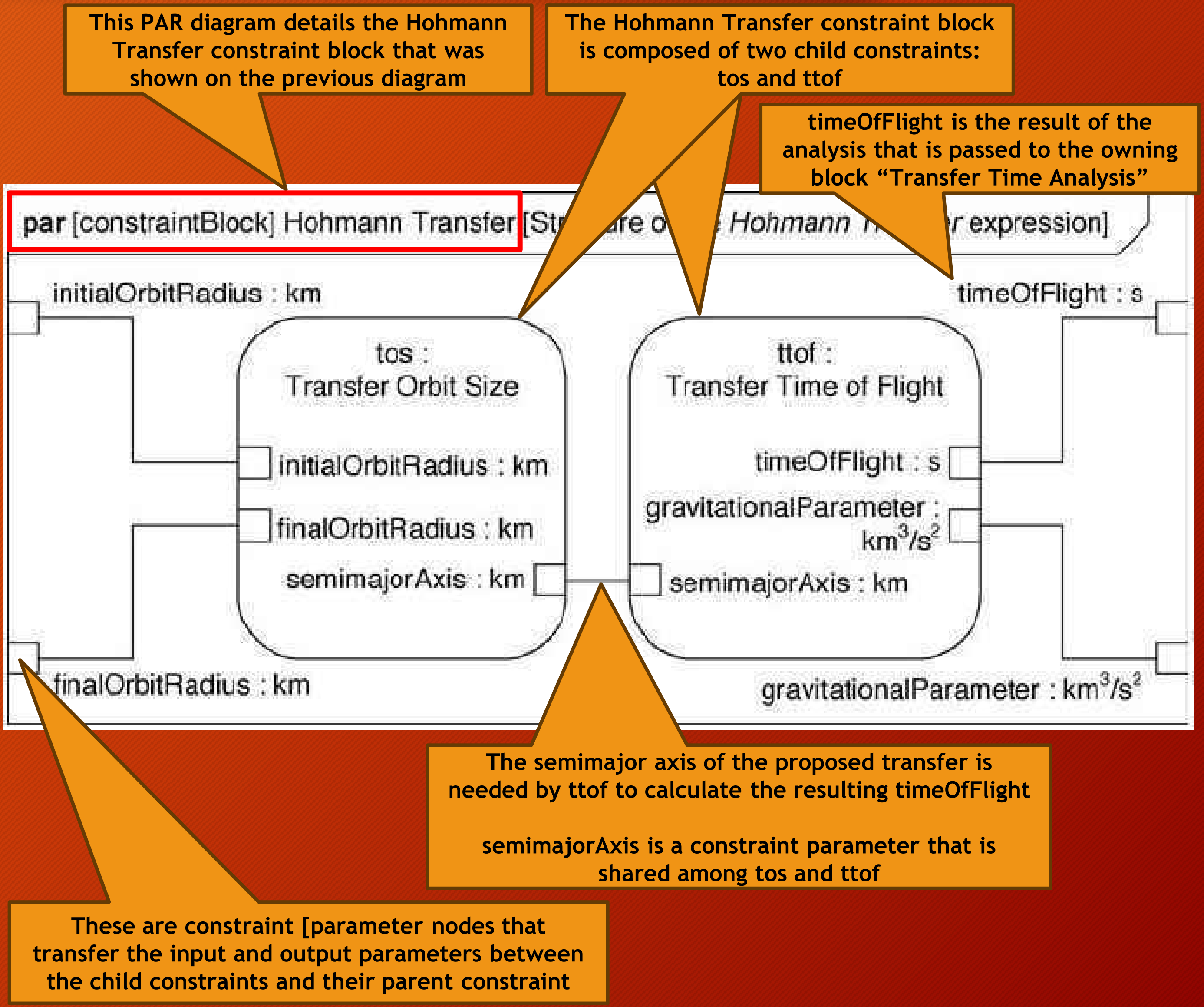
- The Delligatti* example PAR diagrams will be used to identify the syntax and explain the semantics of the diagram
- The diagram meaning will not be clear until we get further into the presentation of the specifics of these examples, but for now it is important to address syntax and semantics
- The frame of the PAR can either be a block whose constraints are being described (as shown in this example)
- Or the frame could represent a parent constraint, whose subordinate children are being described, as in the next slide



* SysML Distilled by Lenny Delligatti

Syntax and Semantics (continued)

- The purpose of the analysis shown in this example is to calculate the time is takes for the Hohmann Transfer to complete
- This PAR diagram shows how the parent constraint Hohmann Transfer is decomposed into its two child constraints tos and ttof
- The computational basis for this analysis can be deduced by observing that
  - initialOrbitRadius and finalOrbitRadius are fed to tos to calculate the two orbits' common semimajor axis
  - semimajorAxis is then passed to ttof which, together with the gravitationalParameter, calculates the resulting timeOfFlight



Delligatti Example Explained

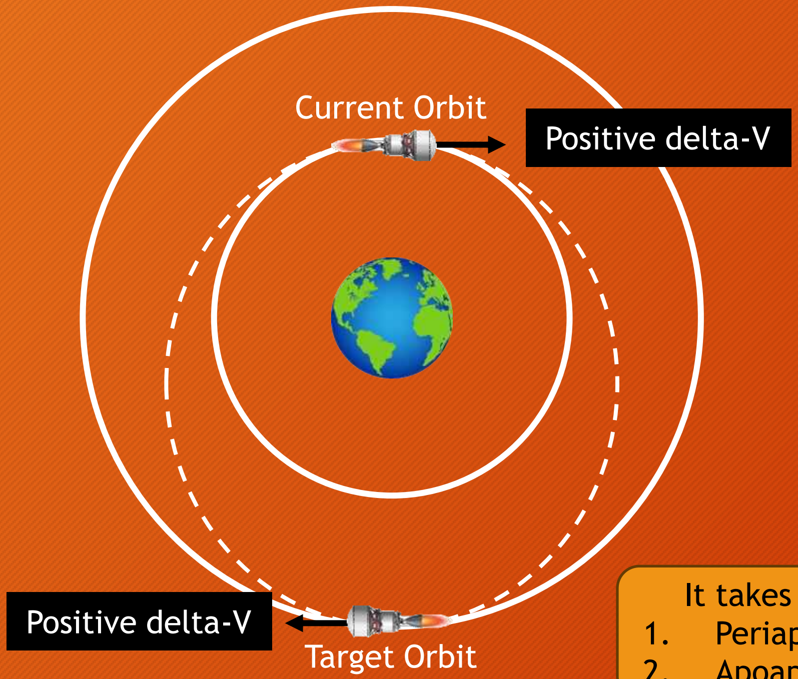
Delligatti Example Explained

- Let us now turn our attention to the parametric analysis example described in the text “SysML Distilled” by Lenny Delligatti
- In this text, a more realistic example of a parametric analysis is given, but requires a bit more detailed explanation in order to be better understood
- This example addresses the subject of orbital mechanics and the transfer of a satellite from one circular orbit to another using a method known as the Hohmann Transfer
- It is not necessary to be well versed in astrodynamics or orbital mechanics in order to follow this example
- The example will be presented by use of textual explanation that does not require such specialized skills

Hohmann Transfer Explained

- The Hohmann Transfer is a simple and efficient way to transfer a satellite from one circular orbit to another
- The Hohmann Transfer can be employed to either
  - Raise the altitude of the circular orbit (rocket burn in direction of velocity)
  - Lower the altitude of the circular orbit (rocket burn against direction of velocity)

Raise the Altitude



It takes two counter-velocity burns to lower the altitude

1. Apoapsis burn – to lower the altitude
2. Periapsis burn – to circularize into the target orbit

The change in velocity required to accomplish these transfers is called “delta-V”

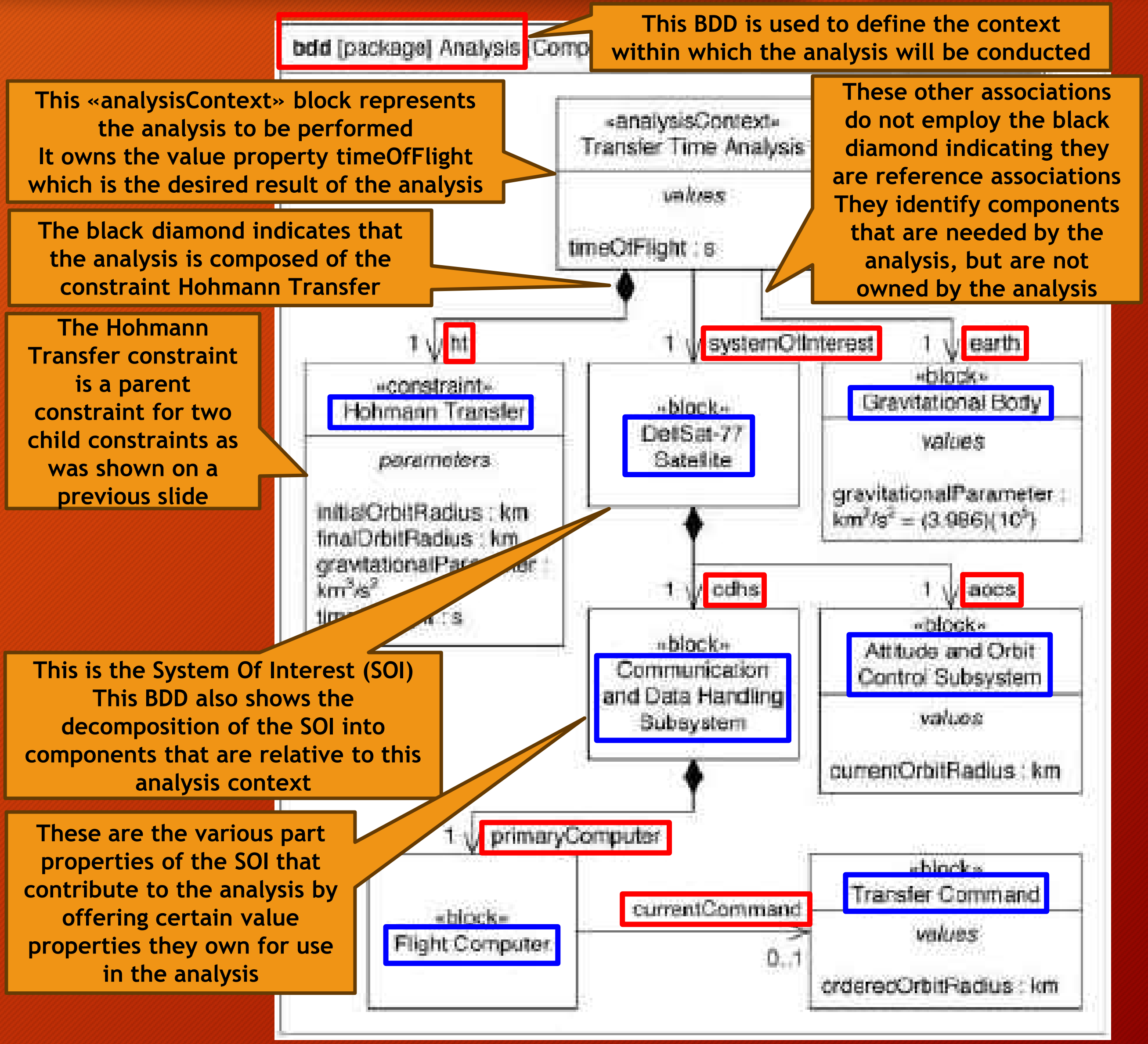
It takes two co-velocity burns to raise the altitude

1. Periapsis burn – to raise the altitude
2. Apoapsis burn – to circularize into the target orbit

Lower the Altitude

Defining the Analysis Context

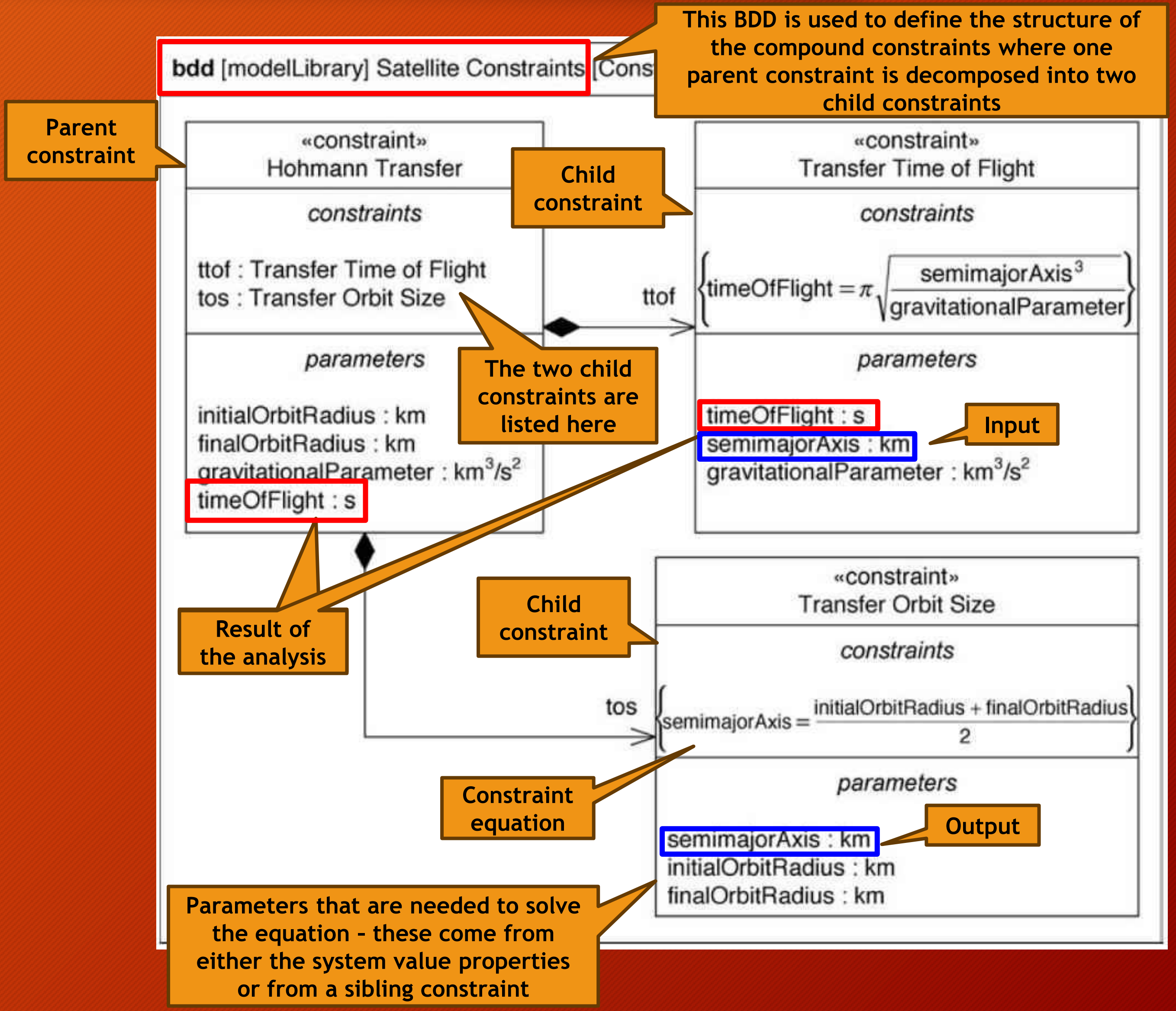
- This BDD is used to define the context of the analysis to be conducted
- It shows two aspects of the analysis
  - It shows the constraints that the analysis owns (here it is just one parent constraint)
  - It also shows the decomposition of the system in order to identify the value properties that are owned by various parts of the system
- Notice that the Transfer Time Analysis block owns the value property timeOfFlight, which will end up being the result of the analysis



Notice that all blocks subordinate to the analysis have role names and type names

Defining the Constraint Structure

- This BDD is used to define the structure of the constraints used in the analysis
- In this analysis, there exists a parent constraint which decomposes into two child constraints
- This BDD shows the decomposition of the parent constraint
- Note that the resulting timeOfFlight, which is shown to be owned by the analysis context in the previous slide, is computed by ttof, and sent to constraint Hohmann Transfer where it can then be passed to the analysis context value parameter



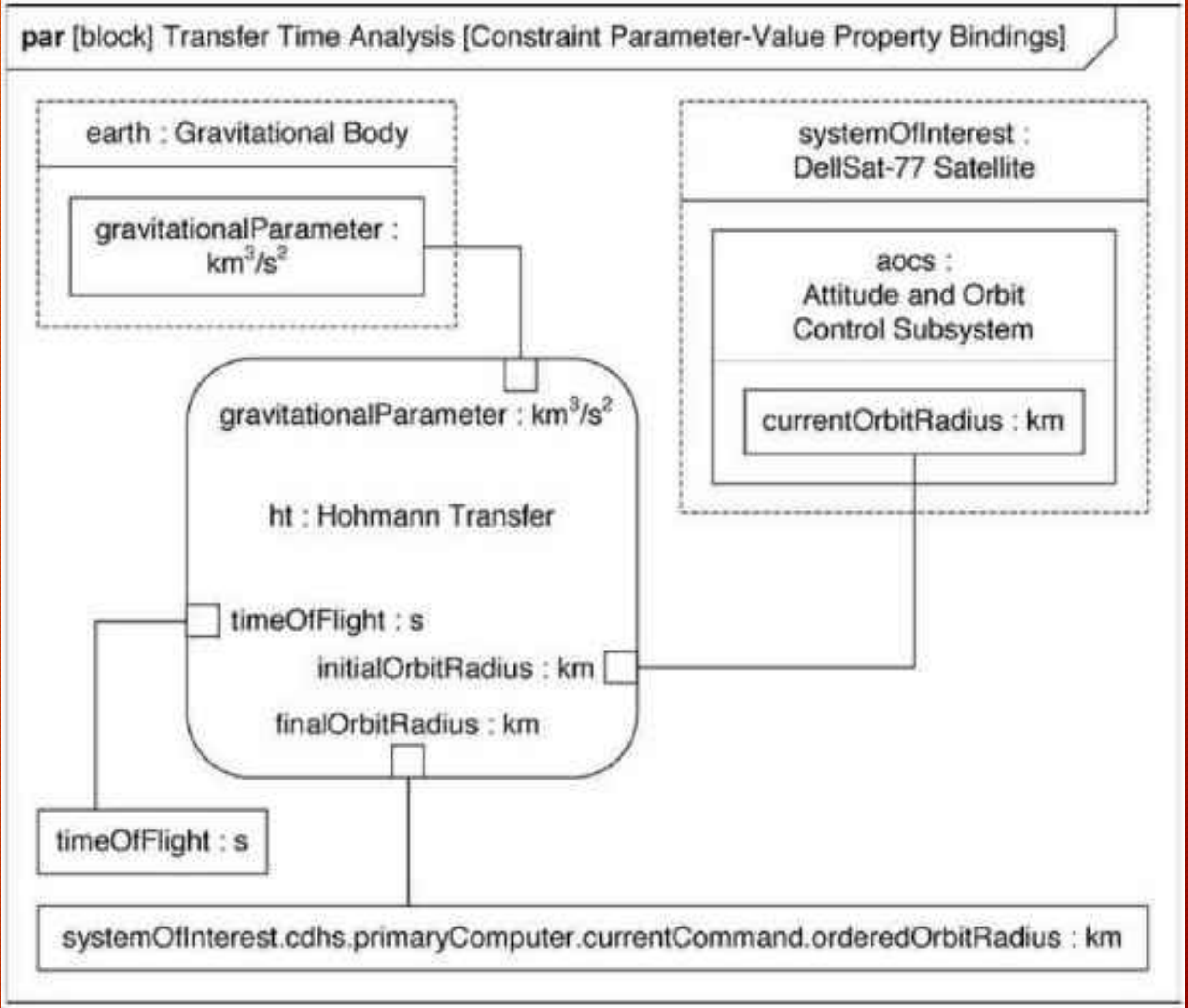
Binding Constraint Parameters to System Value Properties

- The inputs to the analysis come from various elements within the context of the analysis

Input Value Property	Owned By	Passed To Input Parameter
gravitationalParameter	earth	gravitationalParameter
currentOrbitRadius	aocs	initialOrbitRadius
finalOrbitRadius	currentCommand	finalOrbitRadius

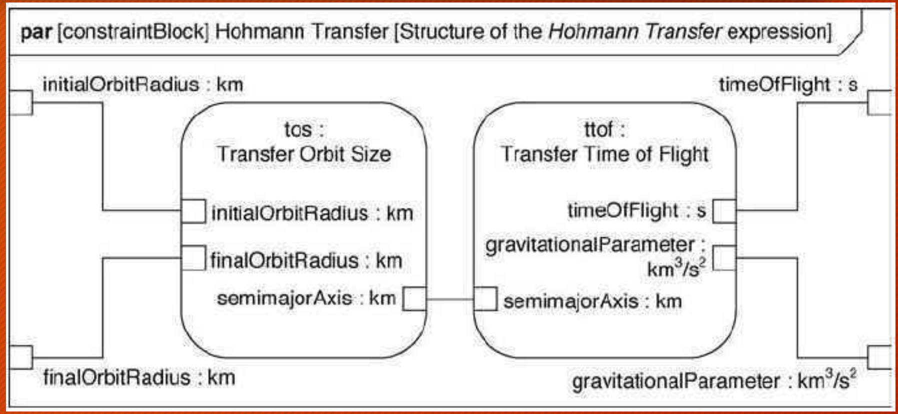
- The output of the analysis is passed to the Analysis itself

Output Parameter	Passed To Value Property	Owned By
timeOfFlight	timeOfFlight	Transfer Time Analysis



Binding Constraint Parameters to other Constraint Parameters

- This PAR diagram shows how the parent's constraint parameters are passed to the child constraints where they Are used in the constraint equations to eventually produce the analysis result timeOfFlight
- The result is passed by the child constraint Transfer Time of Flight to the parent constraint where it is then passed to the value property owned by the Time Transfer Analysis



Parent Input Parameter	Owned By	Passed To Input Parameter	Owned By
initialOrbitRadius	Hohmann Transfer	initialOrbitRadius	Transfer Orbit Size
fianlOrbitRadius	Hohmann Transfer	finalOrbitRadius	Transfer Orbit Size
semimajorAxis	Transfer Orbit Size	semimajorAxis	Transfer Time of Flight
gravitationalParameter	Hohmann Transfer	gravitationalParameter	Transfer Time of Flight

Child Output Parameter	Owned By	Passed To Parent Output Parameter	Owned By
timeOfFlight	ttof	timeOfFlight	Hohmann Transfer

Simple Refrigerator Motor Example

This simple example follows, more-or-less, the same approach as the Delligatti example

Simple Refrigerator Motor Constraint Equations

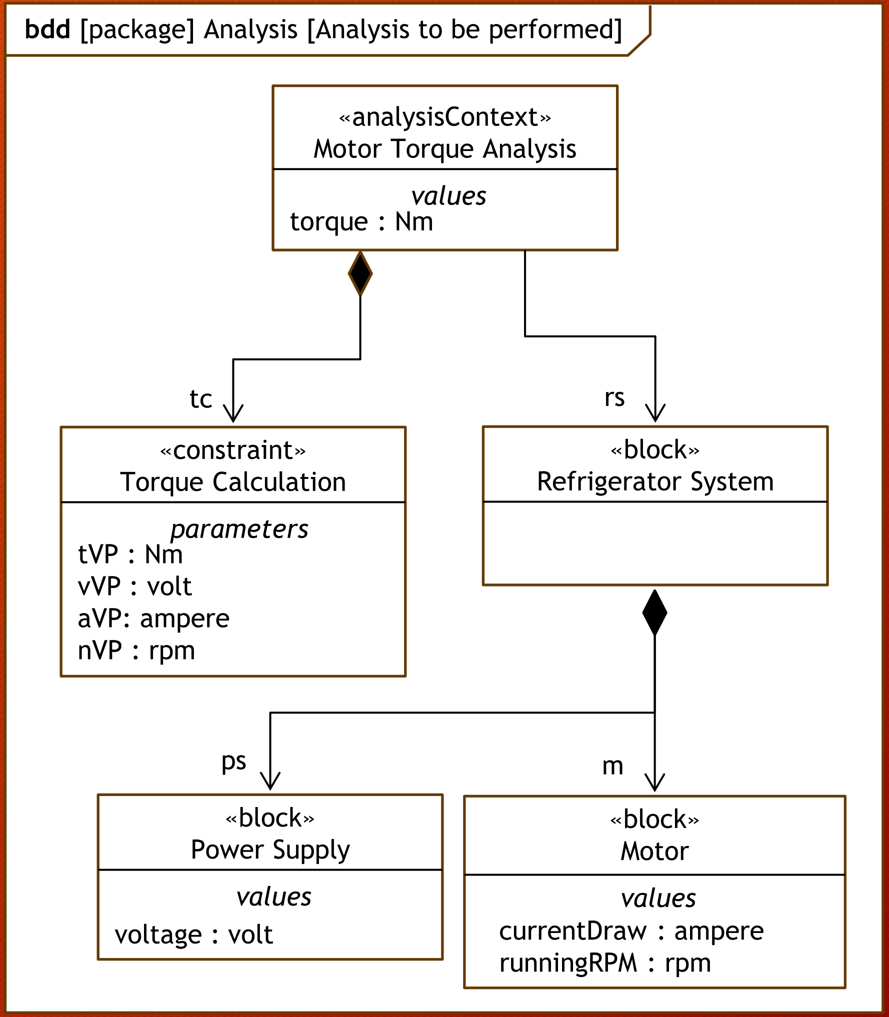
- Let the torque of the refrigerator compressor motor be expressed as:
  - T = P * 9.549 / n
  - where
    - T = motor torque (Nm)
    - P = motor power (watt)
    - n = revolutions per minute (rpm)
- Power can be expressed as:
  - P = V * I
  - Where
    - V = motor voltage (volt)
    - I = motor current (ampere)
- In this very simple representation of the torque produced by a motor, the one parameter that is shared among the two constraints is Power
- This is because he have decomposed the entire problem into two equations that make it easier to understand the problem and to better deal with it organizationally

This example is purposely kept extremely simple in order to focus on the logistical tasks at hand in:

1. Defining the blocks and other elements that go into the analysis
2. Describing the parametric relationships that exist among the various parameters and value properties involved in the analysis

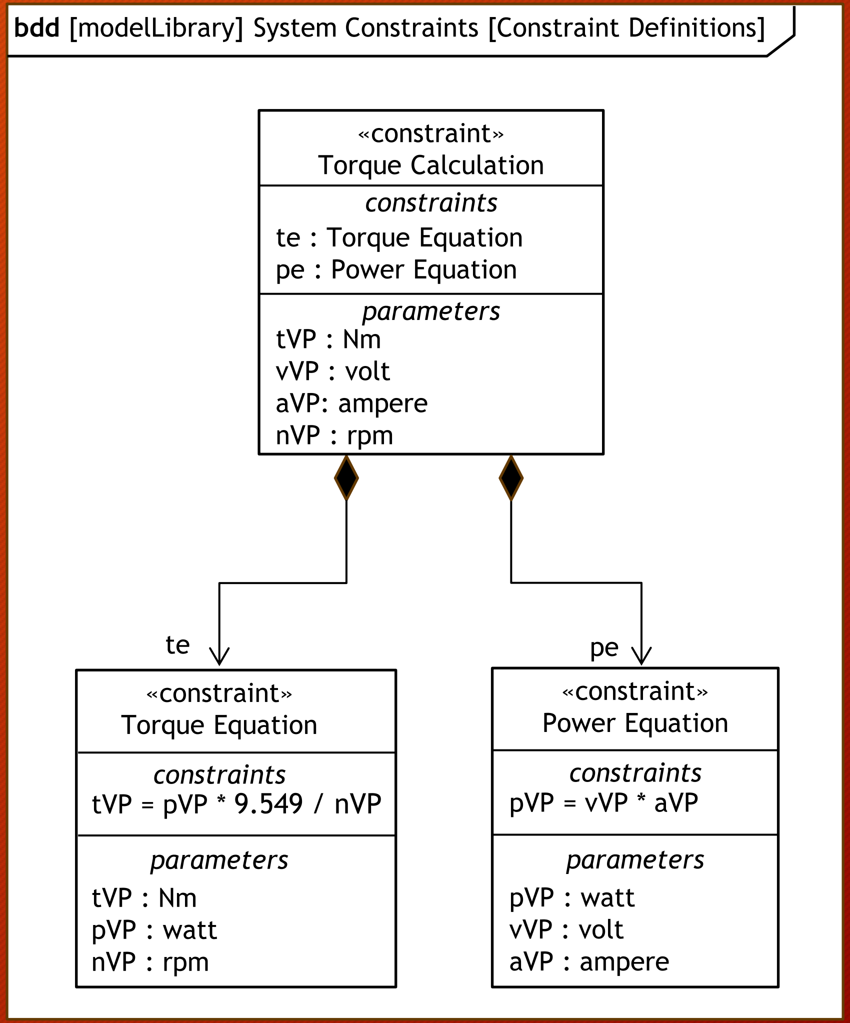
Definition of the Analysis Context of the Parametric Analysis

- You should be aware that the focus of the effort to define these parametric relationships is to support a parametric analysis that helps to establish the potential performance of the system, based on values applied to the constraint parameters
- This "analysis" is defined in a BDD to be composed of a set of constraints, and to reference one or more system value properties that describe the values of the parametric inputs that go into the analysis
- In the example to the right, the BDD is defining the "analysisContext" within which the constraints and value properties will be applied



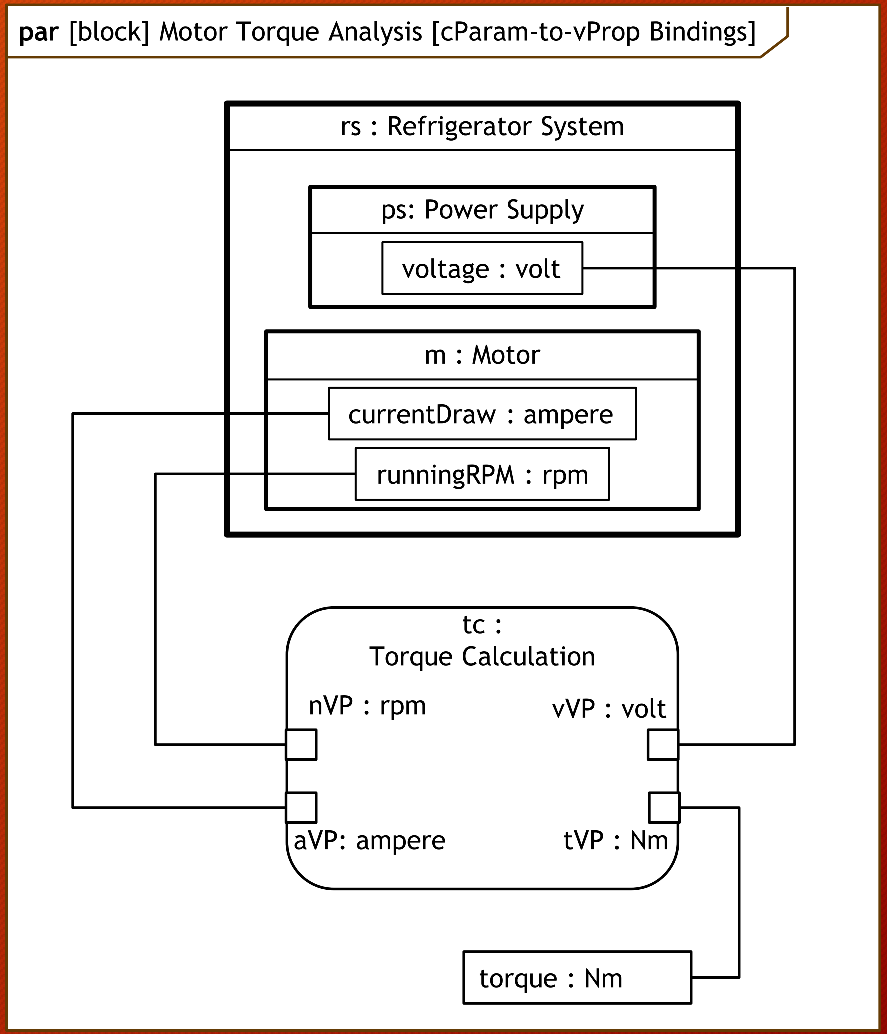
Definition of the Decomposed Abstract Constraint

- Since we decomposed the larger, more complex single constraint equation into two smaller, easier to understand equations, then we need to show that decomposition of a parent constraint into two child constraints
- The place to do that is in a BDD
- BDDs are made to show the decomposition of one block into smaller, easier to understand composite elements
- This BDD shows how the larger Torque Calculation has been broken up into
  - A Torque Equation
  - And a Power Equation that supports the Torque Equation with the power calculation



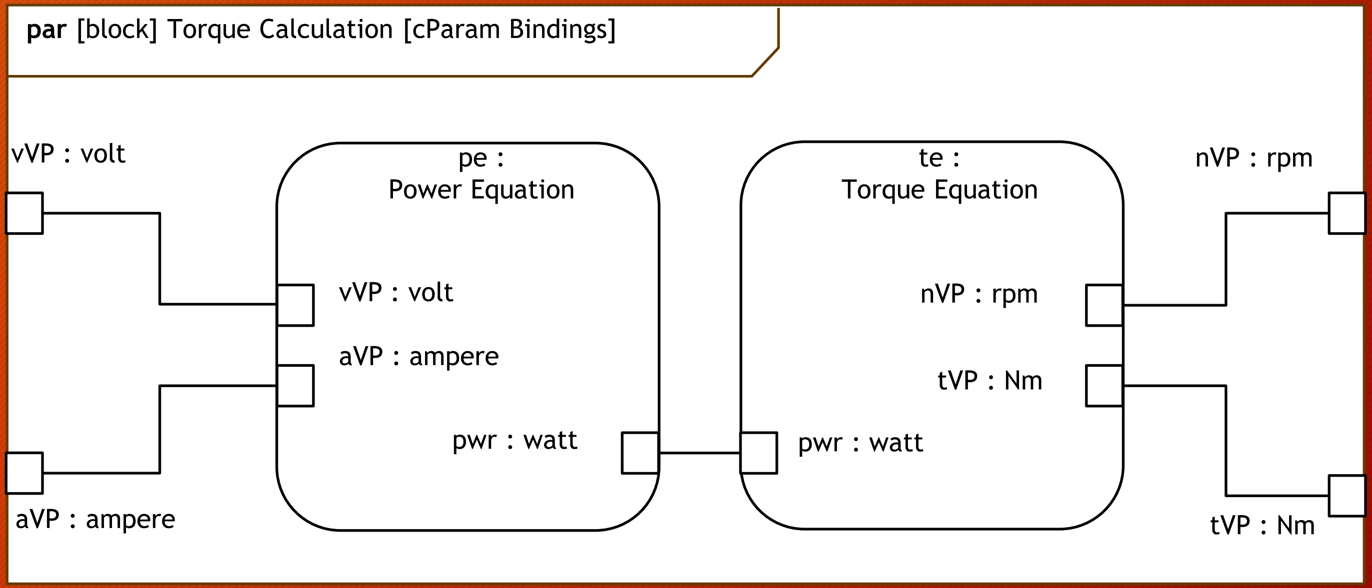
Binding of Constraint Parameters to Value Properties

- This parametric diagram describes the bindings among the parent constraint and the system under analysis and/or the analysis itself
- Here, the frame of the PAR represents the Motor Torque Analysis block, and therefore the value property "torque" can stand alone within that frame
- However, the system-related value properties must be placed into the proper namespace by being shown residing within the owning block(s)
- The constraint parameters of the parent constraint are then bound to these various value properties
- The analysis block (the frame of the PAR diagram) owns the "torque" value property, since determining the torque of the motor is the purpose of the analysis



Binding Constraint Parameters within Parent Constraint Context

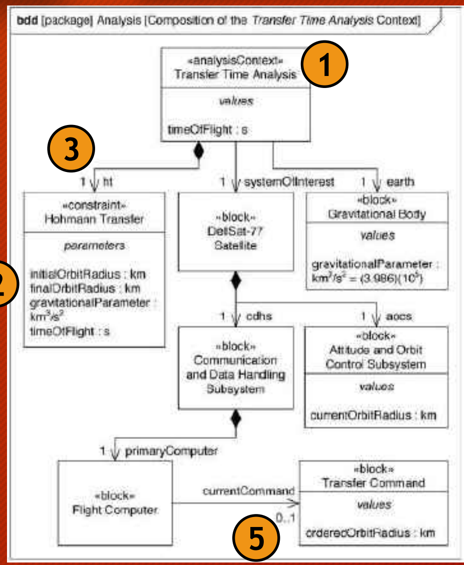
- In this PAR diagram, the relationships among the two child constraints, and those with their parent constraint are shown
- Since the two child constraints share one parameter (“pwr”), the two constraint parameters (one for each child constraint) are bound together
- All other constraint parameters of the child constraints are not bound to each other and therefore must be bound to a constraint parameter of the parent constraint

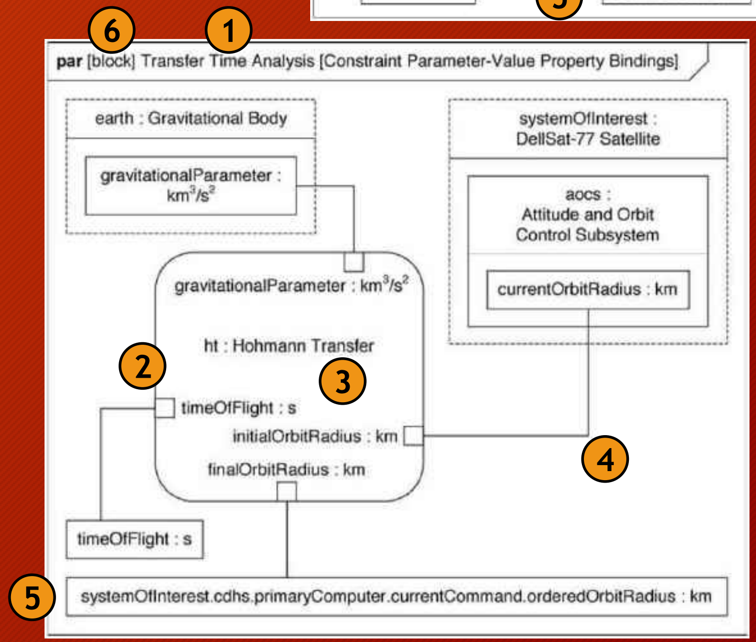


Checklists for Checking the Consistency of the Analysis Definition

Consistency Checking Between Analysis Context BDD and PAR

1. Is the Model Element Name in the PAR header the same as the analysis context block in the BDD?
2. Are all parameters identified in the parent constraint block on the BDD identified as constraint property nodes on the constraint block of the PAR diagram?
3. Is the parent constraint in the PAR diagram identified in the format “role name : constraint type name”?
4. Are all constraint property nodes on the constraint block of the PAR diagram bound to value properties on the PAR diagram?
5. Are all the value properties on the PAR diagram defined on the BDD?
6. Is the Model Element Type of the PAR diagram header of type “block”?





NOTE: Here is an example of how a PAR diagram plays a similar role as an IBD
Here, the PAR diagram shows one valid configuration of the constraint elements defined on the BDD

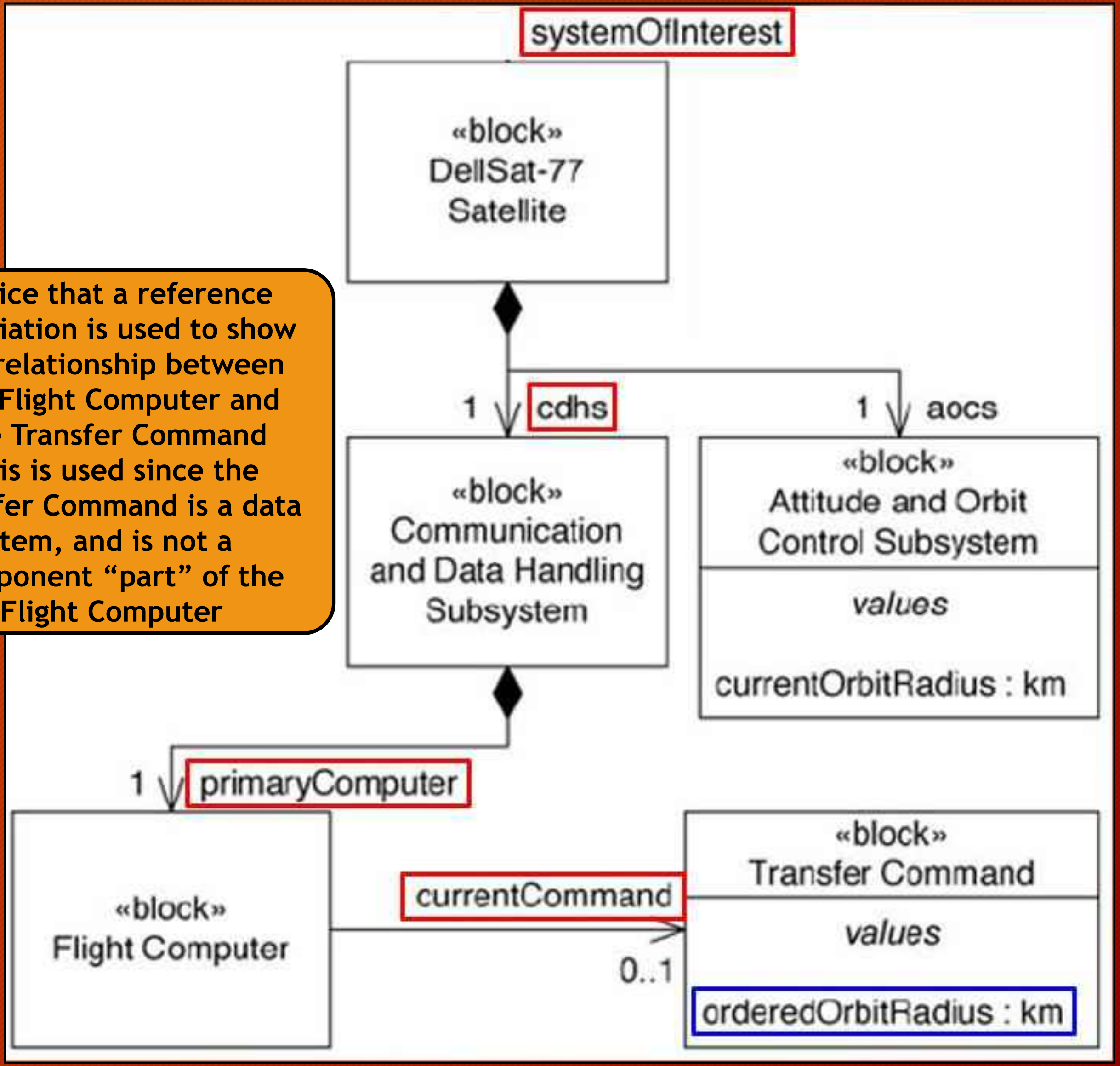
Identifying Elements Using "Dot Notation"

- Notice that in the previous slide, one value property is identified as

systemOfInterest.cdhs.primaryComputer.currentCommand.orderedOrbitRadius : km

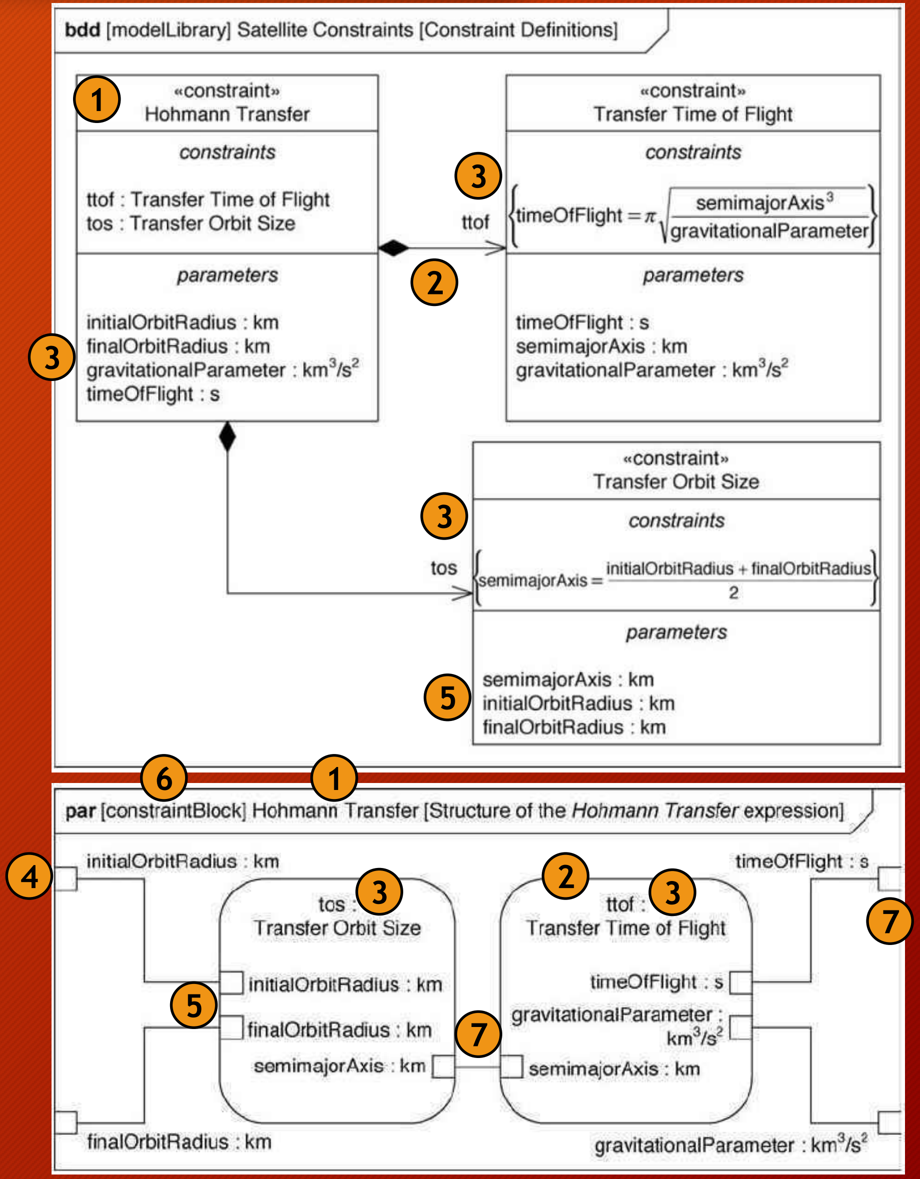
- This reference is making use of dot notation for identifying a system component by tracing the component that owns the value property from the top of the hierarchy (the SoI) down to the component that owns the value property
- The character "." is used to separate the role name of each component in the hierarchical chain
- Compare this specification with the BDD that shows all the components in the chain

Notice that a reference association is used to show the relationship between the Flight Computer and the Transfer Command This is used since the Transfer Command is a data item, and is not a component "part" of the Flight Computer



Consistency Checking Between Constraint Definitions BDD and PAR

1. Is the Model Element Name in the PAR header the same as the parent constraint block in the BDD?
2. Are all constraint blocks identified on the BDD as component constraints of the parent constraint block shown on the PAR diagram?
3. Are all the child constraints in the PAR diagram identified in the format “role name : constraint type name”?
4. Are all the constraint parameters listed for the parent constraint block of the BDD shown on the frame of the PAR diagram?
5. Are all constraint parameters of the child constraint blocks on the BDD shown as constraint property nodes on the child constraint blocks of the PAR diagram?
6. Is the Model Element Type of the PAR diagram header of type “constraintBlock”?
7. Are all the constraint property nodes on the child constraints in the PAR diagram either bound to a sibling constraint property nodes, or to a constraint parameter node on the frame of the diagram?



NOTE: Here is an example of how a PAR diagram plays a similar role as an IBD
Here, the PAR diagram shows one valid configuration of the constraint elements defined on the BDD

References

References

1. L. Delligatti. (2014). *SysML Distilled, A Brief Guide to the Systems Modeling Language*. Pearson Education, Inc. Upper Saddle River, NJ