System Analysis using SysML Parametrics: Current Tools and Best Practices

Chris Paredis
Model-Based Systems Engineering Center
Georgia Tech
Chris.paredis@me.gatech.edu

www.pslm.gatech.edu/courses
www.omg.org/ocsmp

Acknowledgments

- Software Support
  - No Magic, Inc
  - InterCAX
  - Phoenix Integration
- Collaborators
  - Russell Peak (1)
  - Manas Bajaj (1)
  - Roger Burkhart
  - Sandy Friedenthal
  - Leon McGinnis
- Grad Students / Postdocs
  - Aditya Shah
  - Alek Kerzhner
  - Axel Reichwein
  - Ben Lee
  - Kevin Davies
  - Roxanne Moore
  - Sebastian Herzig
  - Wladimir Schamai

(1) provided much of the materials on ParaMagic

Abstract

- The Systems Modeling Language (OMG SysML) is a general purpose modeling language that allows users to express a broad range of information and knowledge about systems and systems engineering projects: from requirements to functional architectures, and from behavior to structural specifications. An important set of modeling constructs in SysML are *parametric constraints* as used in parametric diagrams. SysML parametrics allow systems engineers not only to define parametric system specifications but also to express analysis models relating measures of effectiveness to system specifications. In this tutorial, the use of paramtrics for analysis purposes will be reviewed in detail. To provide the appropriate context, we will start with a short overview of SysML, followed by a more detailed review of the syntax and semantics of parametric diagrams. We will then discuss different uses of parametrics, and provide best-practice examples, specifically focusing on analysis modeling. Parametric diagrams will also be compared with a new OMG specification for combining SysML and Modelica: The OMG SysML-Modelica Transformation Specification. A live demonstration will be included with the following two tools: The MagicDraw SysML Tool with the ParaMagic plug-in, and an open-source plug-in for solving parametric constraint models in ModelCenter.

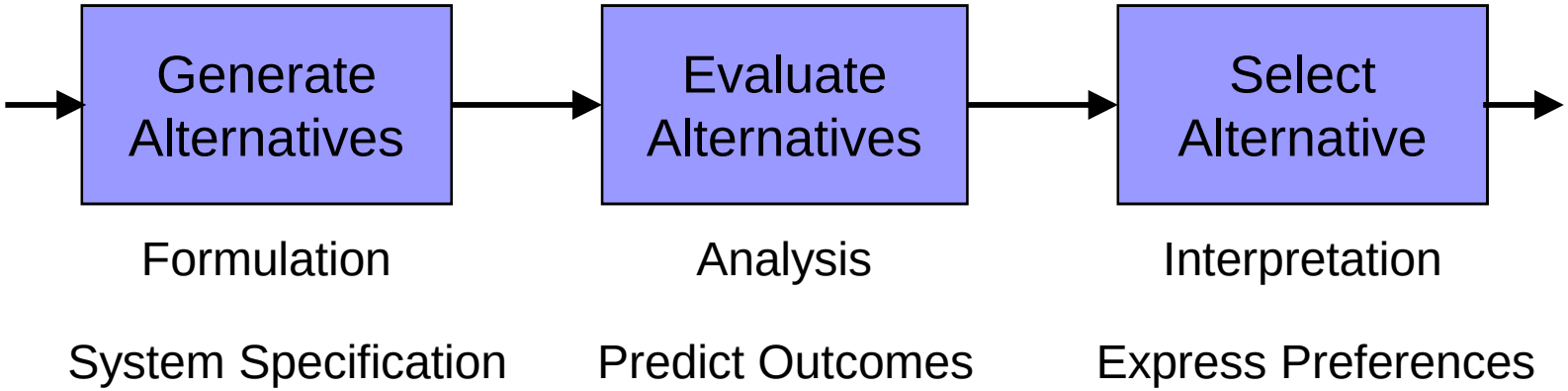
Outline

- Short overview of SysML
- SysML Parametrics: Syntax and Semantics
- Best practices in Paramatrics Modeling
  - Margin analysis – SimpleSat Satellite Example
  - Analysis context
  - Parameter aggregation
- Tools for solving Parametrics
  - ParaMagic
  - ModelCenter
- Parametrics versus SysML4Modelica
- Summary

Coffee Break

Systems Engineering as Decision Making

- Our perspective:
  - Design involves deciding on the most preferred solution alternative
  - The most preferred alternative is the one that leads to the most preferred outcomes



We need a language to express this information and knowledge

What is SysML?

- The Systems Modeling Language (OMG SysML™) is a visual, general purpose modeling language

- **Is a** modeling language that provides
  - Semantics = meaning
  - Notation = representation of meaning

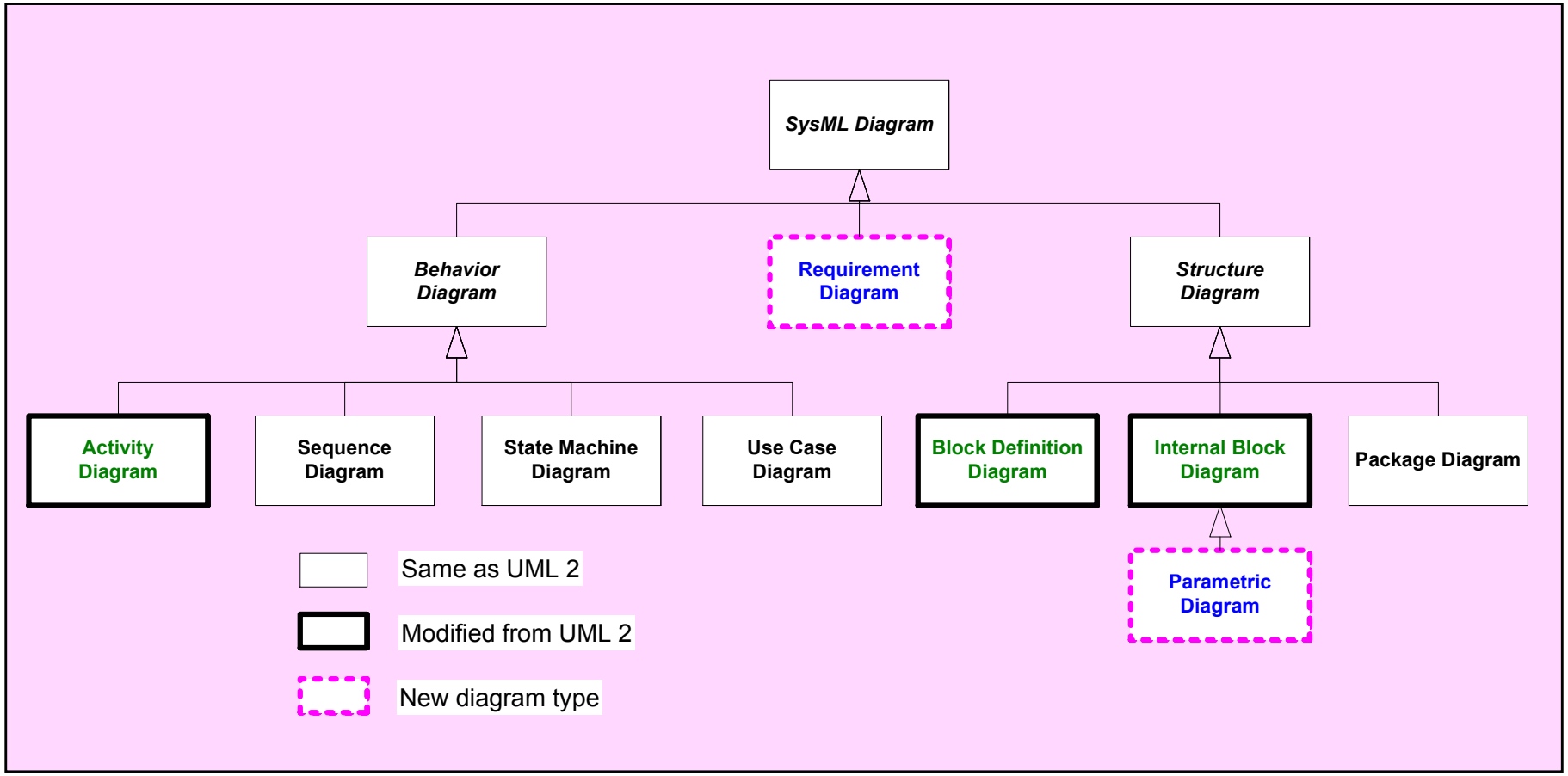
- **Is not a** methodology or a tool
  - SysML is methodology and tool independent

- Developed by the Object Management Group to support Model-Based Systems Engineering

What Can be Expressed in SysML?

- SysML is a language to express the information and knowledge generated and processed during the application of a systems development methodology
  - Specification
  - Analysis
  - Design
  - Verification
  - Validation
  - Hardware
  - Software
  - Data
  - Personnel
  - Procedures
  - Facilities

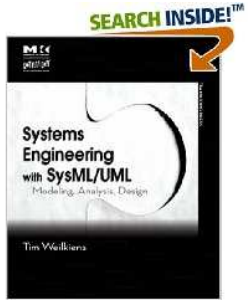
SysML Diagram Taxonomy

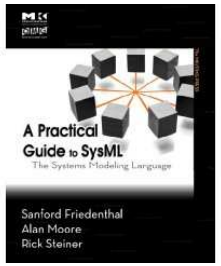


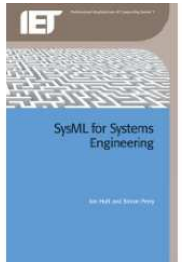
Think of SysML as an integrated collection of languages…

Some History…

- Specification: http://www.omg.org/spec/SysML/
  - v1.0: 2007-09
  - v1.1: 2008-11
  - v1.2: 2010-06
  - v2.x: RFI preparation workshop - 2008-12
- Strong vendor support
  - MagicDraw (No Magic), Artisan Studio (Atego), Enterprise Architect (Sparx Systems), Rhapsody (IBM),…
- Good learning infrastructure
  - Books, short courses, academic courses, INCOSE/OMG tutorial, public examples, etc.
- OMG Certified Systems Modeling Professional
  - http://www.omg.org/ocsmp/







Model-Based Systems Engineering in Industry

- Actively used in most large companies in Aerospace, Defense, Automotive:
  - In a recent SysML survey, 45 companies participated:

Space Systems:	23%
Aircraft:	20%
Defense:	20%
Automotive:	7%
Other:	30%

- No longer small pilot studies!

Project Duration	
1 mo – 1 year:	20%
1 year – 3 years:	35%
> 3 years:	45%

Project Size	
< 10 people:	28%
10 – 100:	40%
100 – 1000:	22%
> 1000 people:	10%

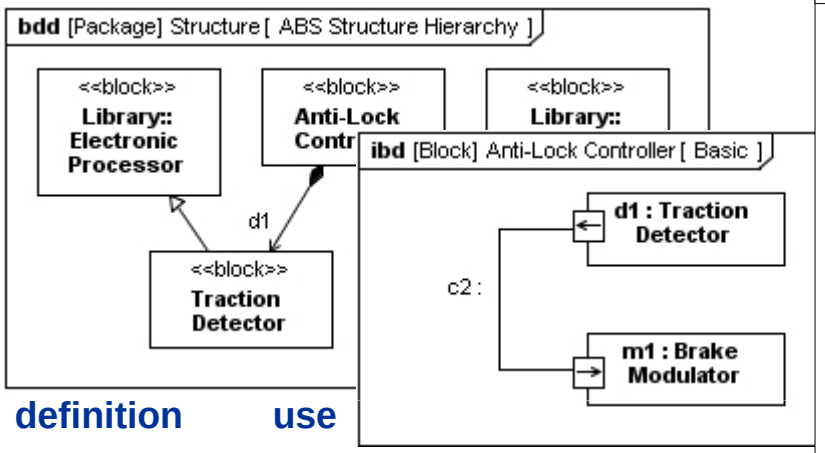
- MBSE is becoming part of day-to-day engineering practice

(Data Source: Robert Cloutier — http://www.omg.org/cgi-bin/doc?syseng/2009-12-04)

What is SysML?

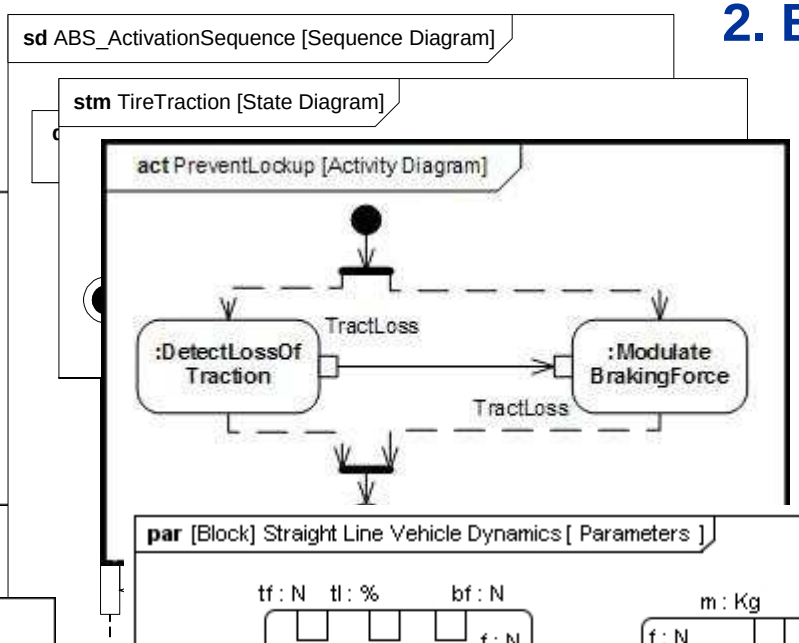
(www.omgsysml.org)

1. Structure



definition **use**

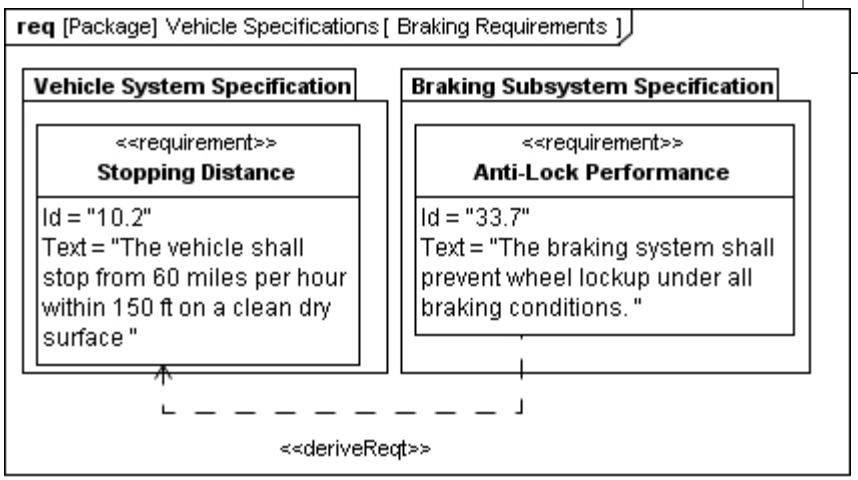
2. Behavior



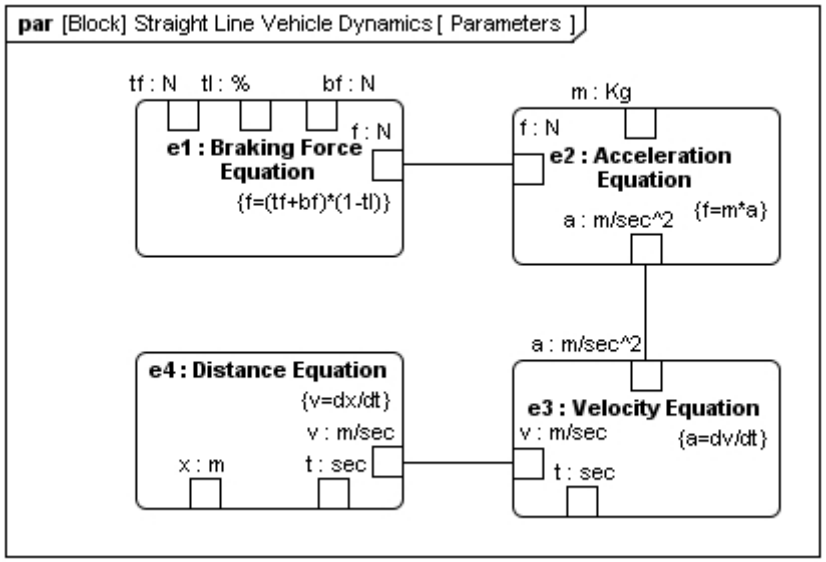
interaction

state machine

activity/ function



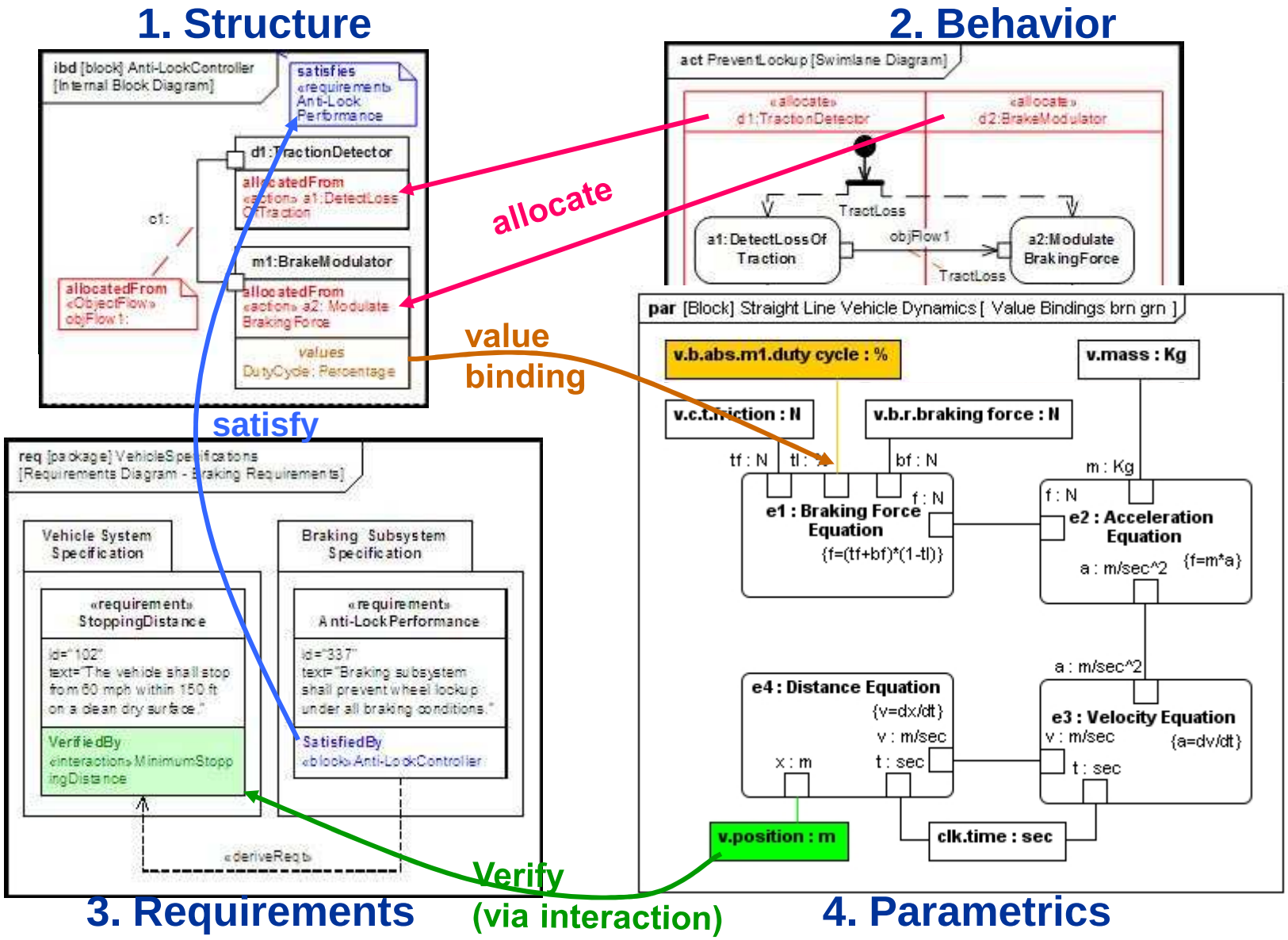
3. Requirements



4. Parametrics

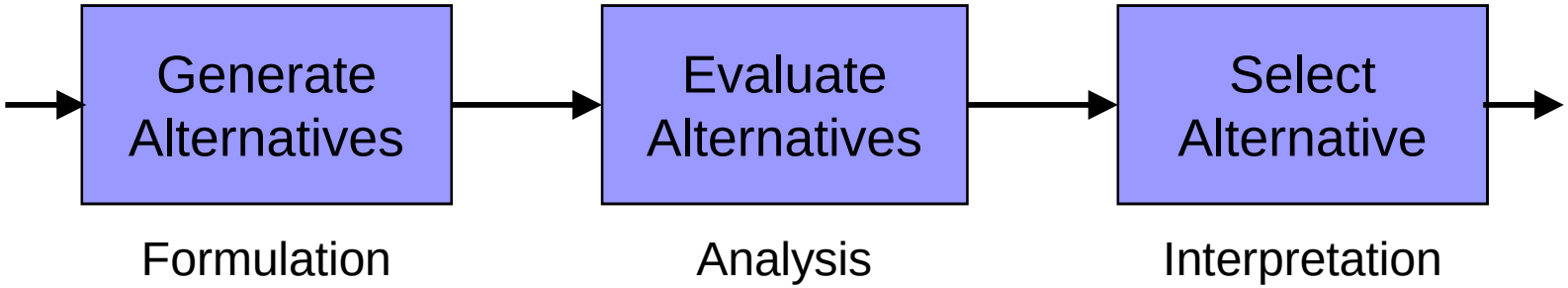
What is SysML?

(www.omgsysml.org)



Systems Engineering as Decision Making

- Our perspective:
  - Design involves deciding on the most preferred solution alternative
  - The most preferred alternative is the one that leads to the most preferred outcomes



BDD, IBD,
ACT, STM, SD, (PAR)
→ Specify the alternatives

PARAMETRIC,
executable (ACT, STM, SD)
(analyses external to SysML)
→ Predict outcomes

Escaping "Spreadsheet Sabotage"

BROWN SMITH WALLACE
CONSULTING GROUP
A MEASURABLE DIFFERENCE™

Are Spreadsheets Sabotaging Your Profits?

Based on a broad study of U.S. corporations, Sanders & Manrodt (2002) determined that almost 90 percent of businesses rely on spreadsheets for their forecasting software. This study also concluded that using commercial software packages for forecasting produced better performance than spreadsheets.

By Steve Epner
Founder Brown Smith Wallace Consulting Group
and Innovator in Residence
at St. Louis University

"The use of spreadsheets to perform complex business functions exposes a business to a number risks..."

"A detailed study by Panko (2000) revealed that many companies are using highly error-prone spreadsheets to perform numerous functions. According to research over a long period of time, Panko has proven that spreadsheet errors are common and result in meaningful, harmful business impacts."

"According to Panko (2005), real life audits found errors in 94 percent of the spreadsheets examined. In summary, it is fair to say that the use of spreadsheets to make business decisions is potentially dangerous."

Steve Epner (2007) Are Spreadsheets Sabotaging Your Profits? Research Note, Brown Smith Wallace Consulting Group St. Louis, www.bswllc.com. available here: http://www.whitakerit.com/documents/Are-Spreadsheets-Sabotaging-Your-Profits.pdf

Discussion

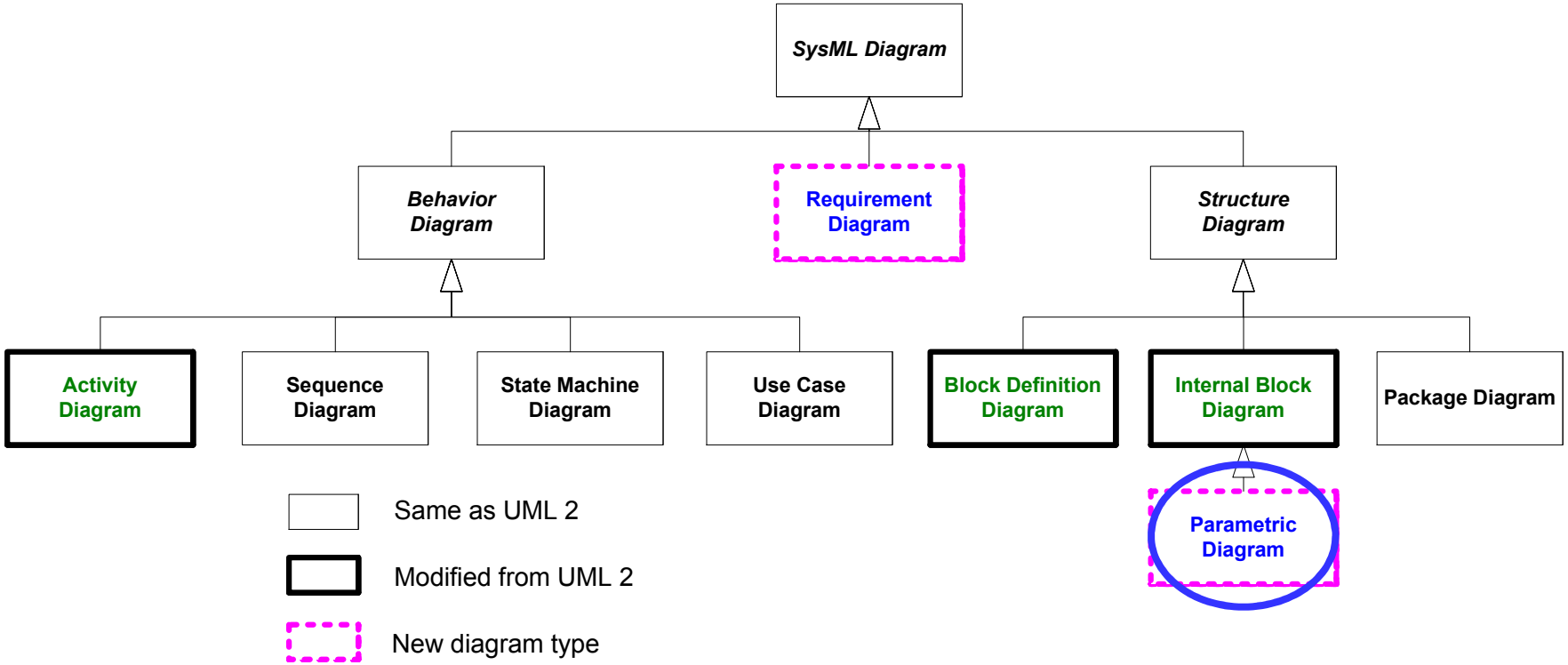
- What are the advantages of representing Systems Engineering in a (formal) model?

Outline

- Short overview of SysML
- SysML Parametrics: Syntax and Semantics
- Best practices in Paramatrics Modeling
  - Margin analysis – SimpleSat Satellite Example
  - Analysis context
  - Parameter aggregation
- Tools for solving Parametrics
  - ParaMagic
  - ModelCenter
- Parametrics versus SysML4Modelica
- Summary

Coffee Break

Parametric Diagrams in SysML



Expressing Constraints in SysML

- Need to represent mathematical relationships or constraints in SysML

- Equation = mathematical constraint (or relationship) between parameters (or variables)

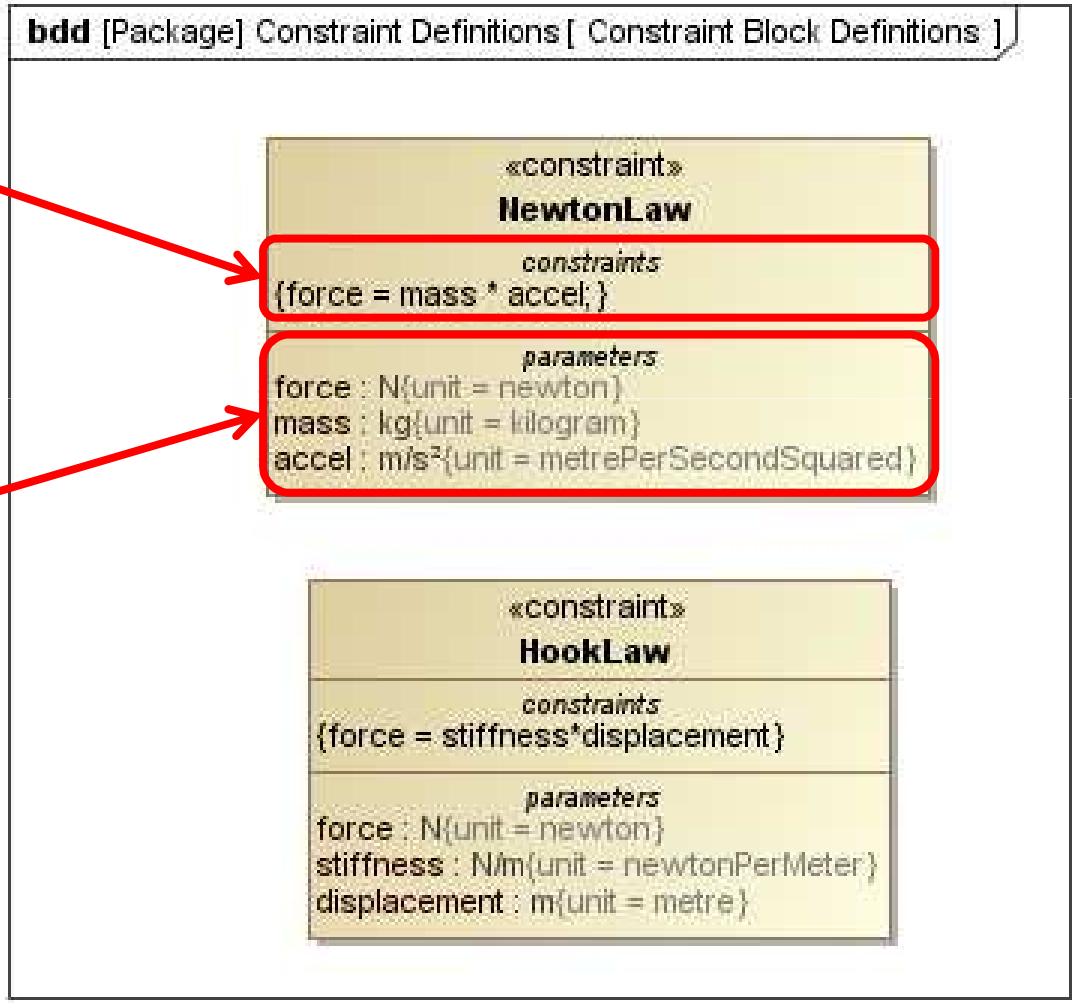
- Equation versus assignment:
  - What is the difference between:

$$F = m*a$$
$$m*a = F$$
$$F - m*a = 0$$
$$F / m = a$$

In SysML constraints are modeled ***declaratively***:
- state what must be true
- not how to compute it

Defining a Constraint in SysML

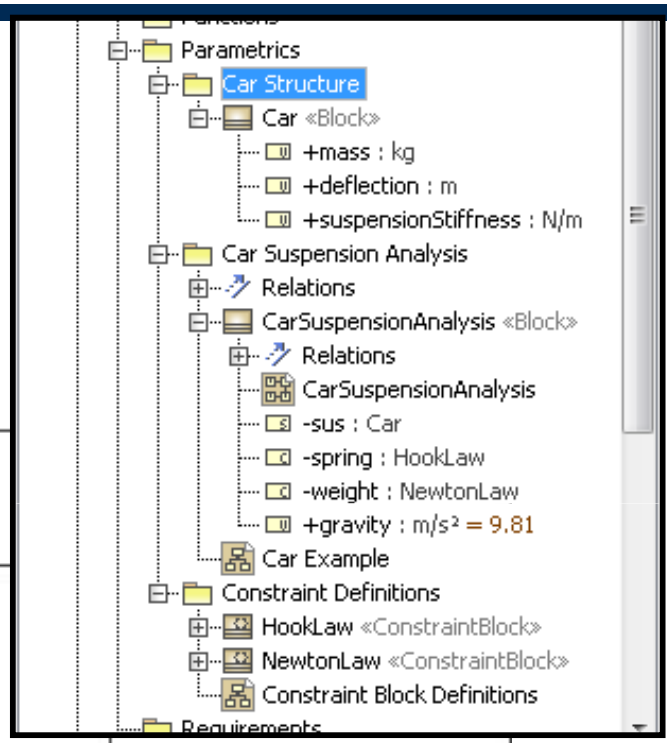
- Constraint Block
  - Constraint (optional: specify language in which constraint is expressed)
  - Constraint parameters

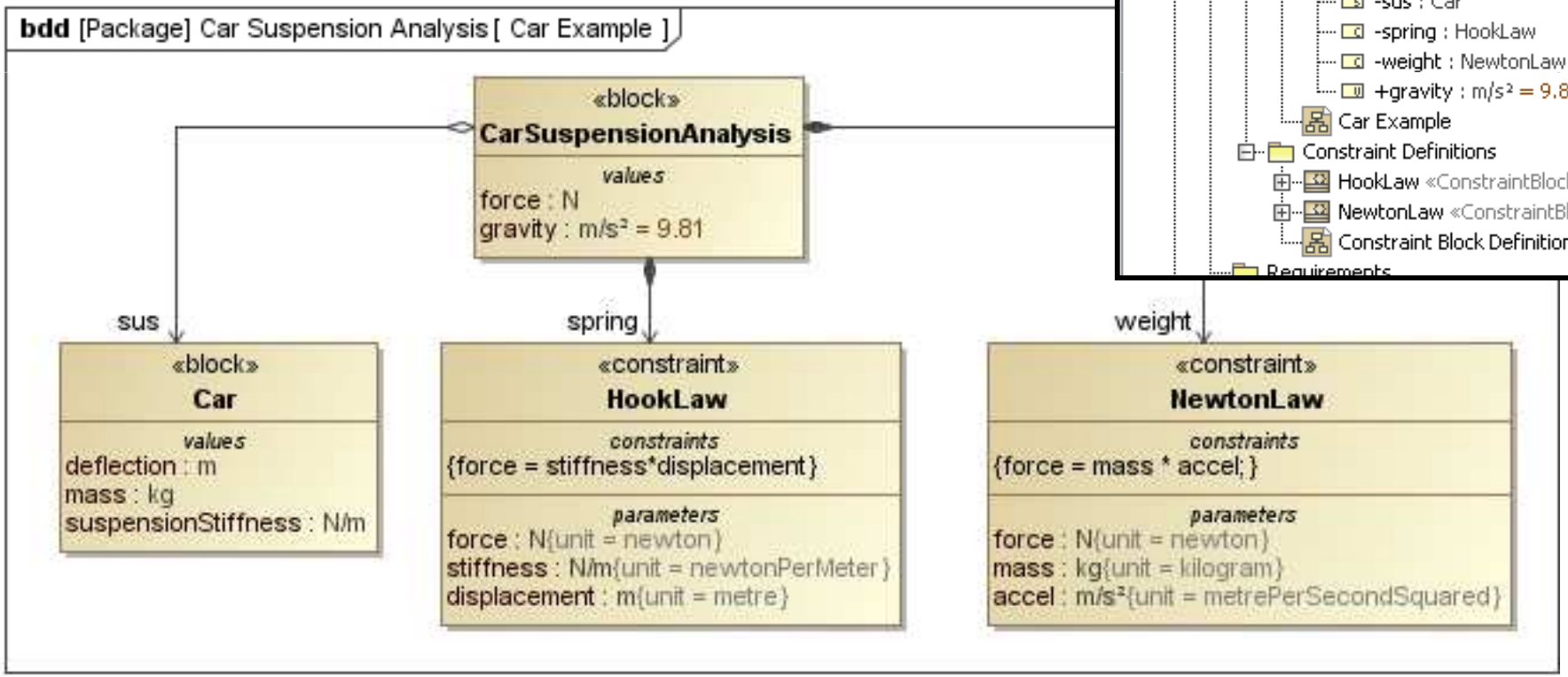


- Compare to Block:
  - Block ~ Constraint Block
  - ValueProperty ~ Constraint Parameter

Using a Constraint Block: Constraint Property

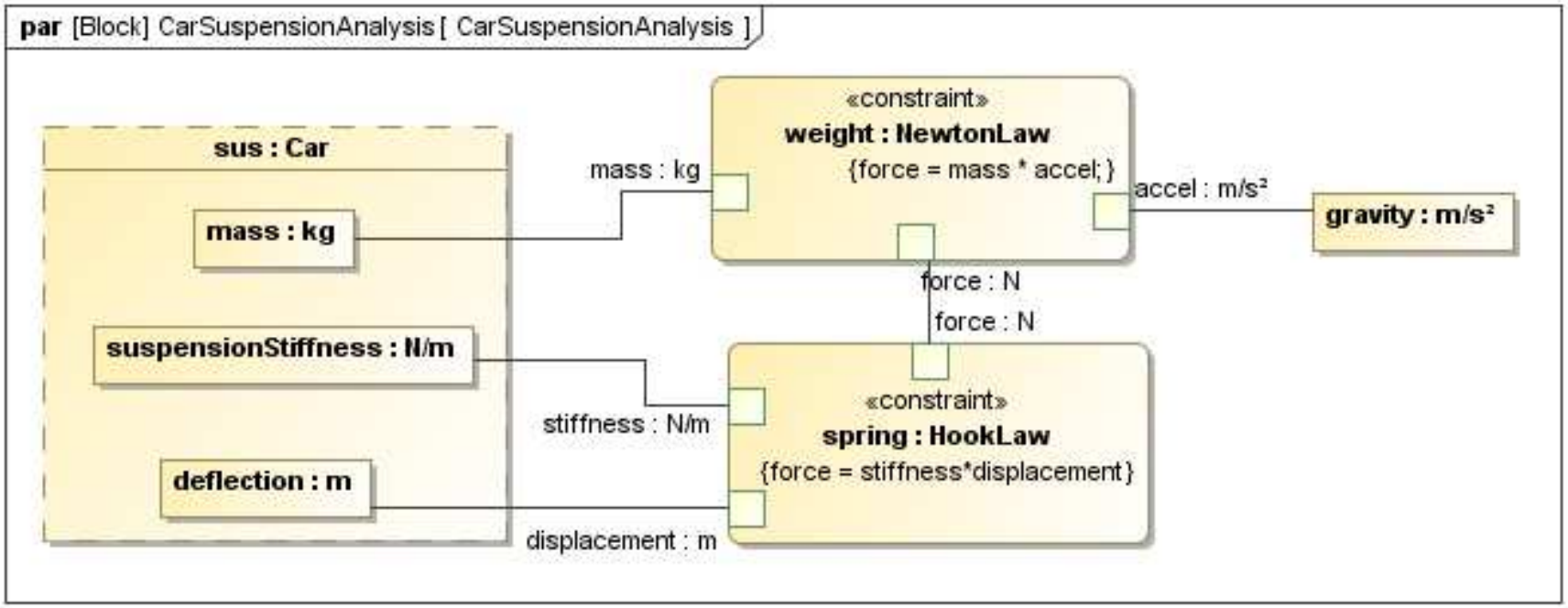
- Constraint Property is owned by a Block
- Constraint Property is typed to a Constraint Block





Using a Constraint Block: Parametric Diagram

- Constraint Property constrains the value properties of the block that owns it
- The value properties are *bound* to constraint parameters using binding connectors
  - Meaning of binding connector: bound values are equal
  - Be careful: only values that are type compatible can be bound



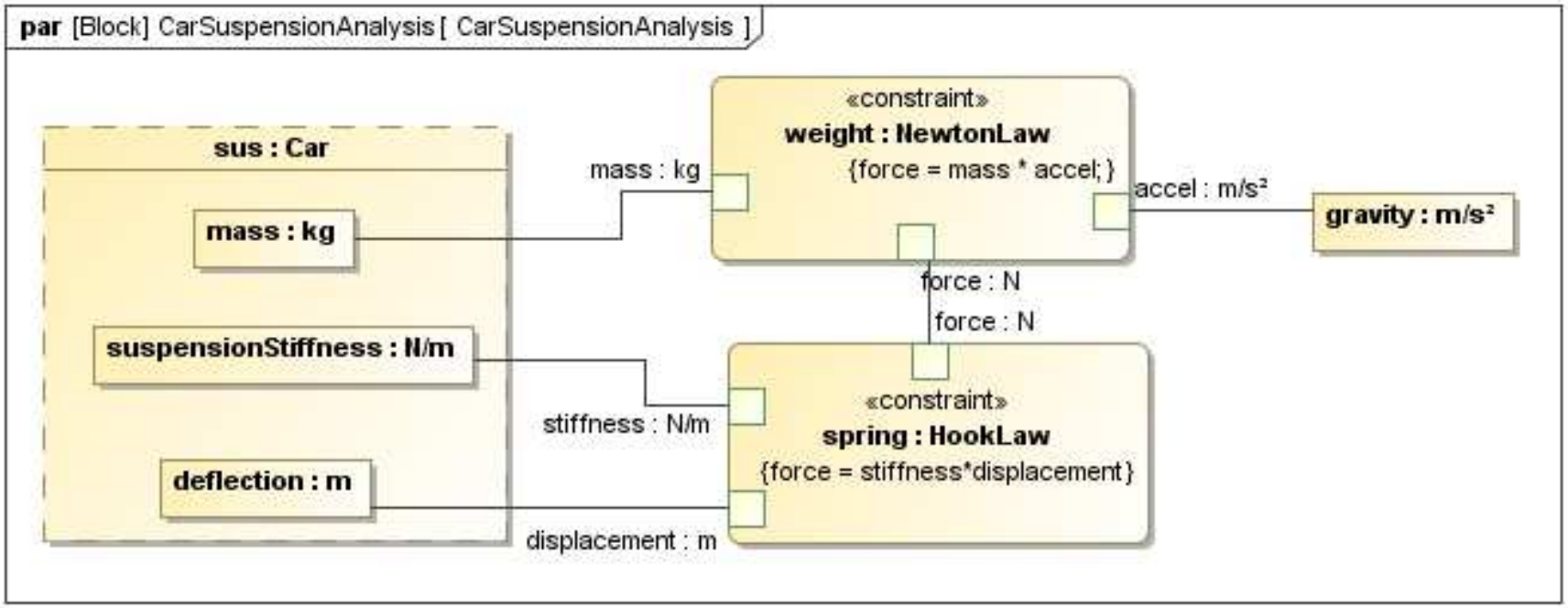
How is a PAR different from an IBD?

PAR

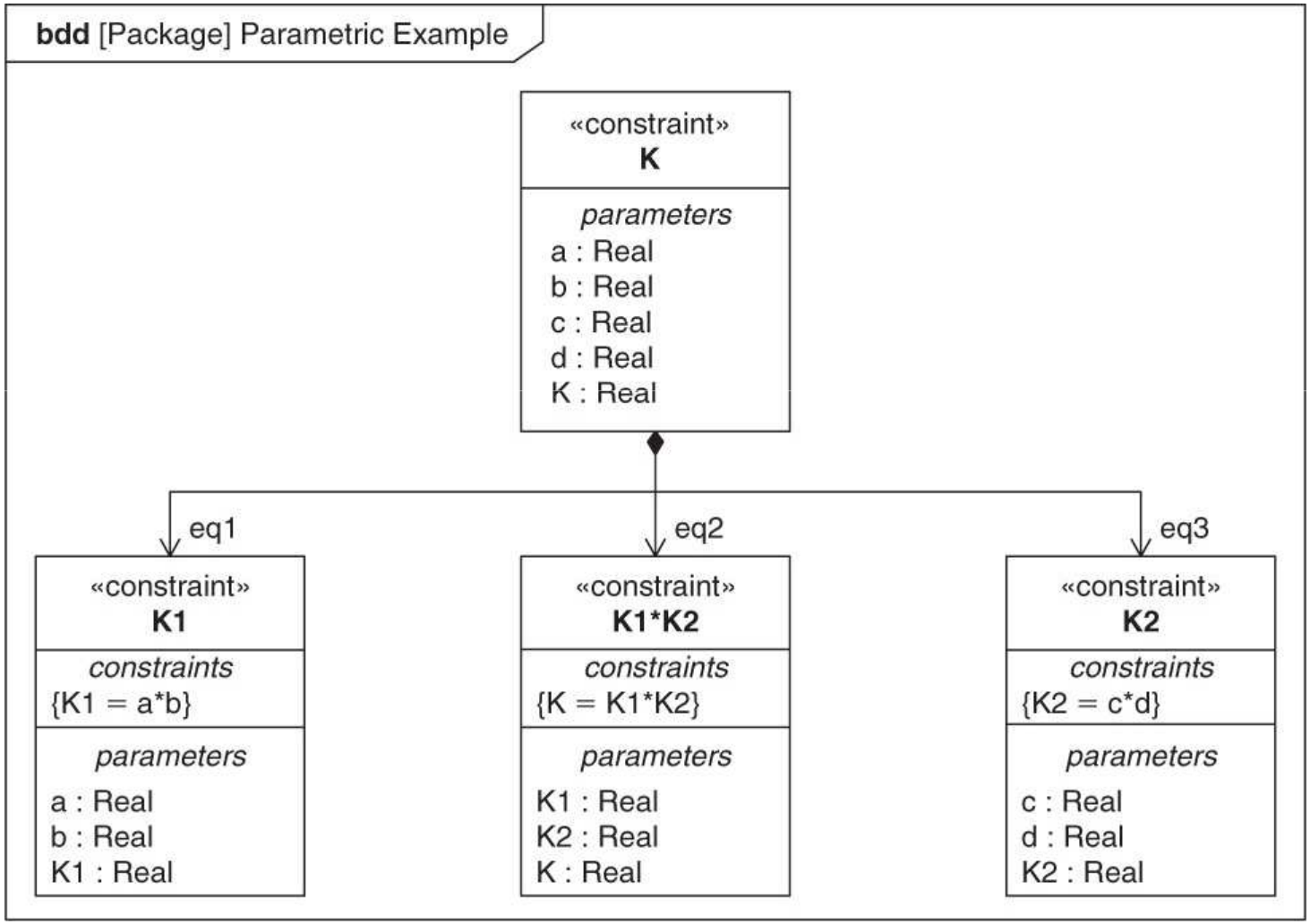
- Connect constraint parameters / value properties
- Binding connectors
- Constraint properties

IBD

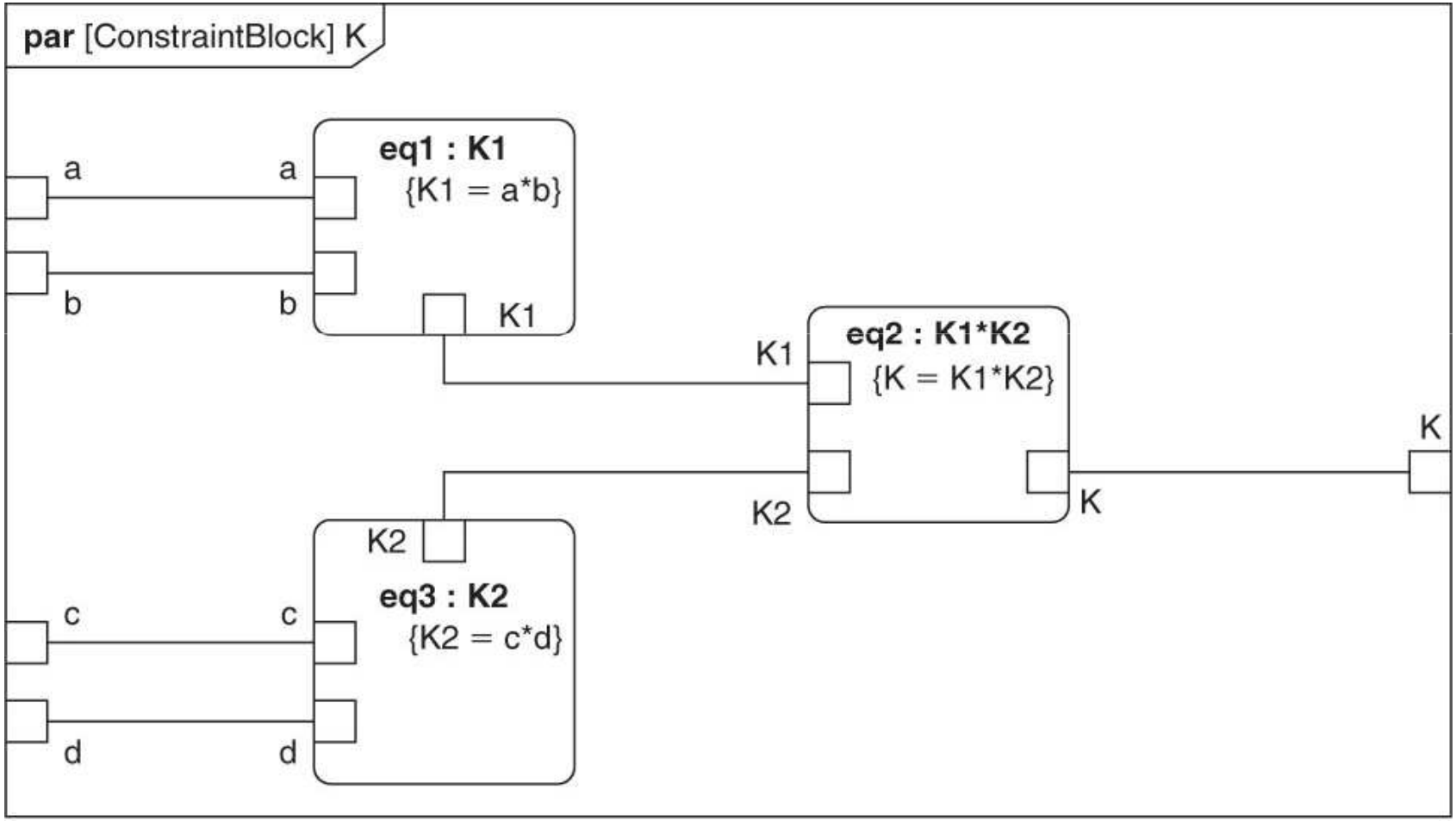
- Connect ports / flow ports / parts
- Connectors
- Part properties



Nested Constraint Blocks



Nested Constraint Blocks



Review of PAR Graphical Notation

Diagram Element	Notation	Description	Section
Constraint Parameter Node	<Name>: <Type>[<Multiplicity>]	A constraint parameter is a special kind of property that is used in the constraint expression of a constraint block. Constraint parameters do not have direction.	7.3
Constraint Property Node	<Name>:<ConstraintBlock>[<Multiplicity>] {<Constraint>} «constraint» <Name>:<ConstraintBlock>[<Multiplicity>] {<Constraint>}	Constraint properties are defined by constraint blocks and used to bind (i.e., connect) parameters. This enables complex systems of equations to be composed from more primitive equations, and for the parameters of the equations to explicilty constrain properties of blocks.	7.4
Value Binding Path	<Multiplicity> <Multiplicity> <Multiplicity> «equal» <Multiplicity>	Binding connectors connect constraint parameters to each other and to value properties. They express an equality relationship between their bound elements.	7.5

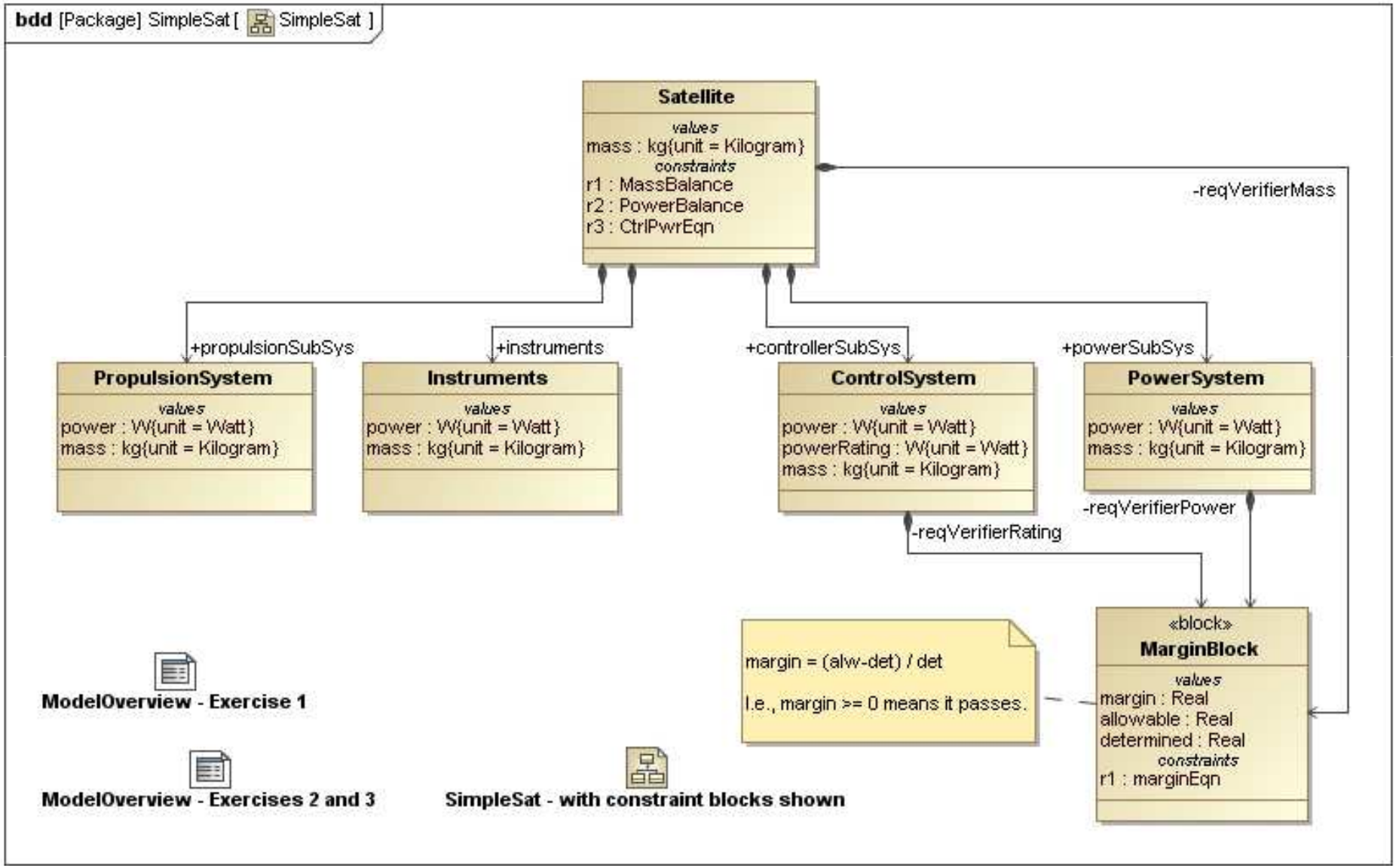
Outline

- Short overview of SysML
- SysML Parametrics: Syntax and Semantics
- Best practices in Paramatrics Modeling
  - Margin analysis – SimpleSat Satellite Example
  - Analysis context
  - Parameter aggregation
- Tools for solving Parametrics
  - ParaMagic
  - ModelCenter
- Parametrics versus SysML4Modelica
- Summary

Coffee Break

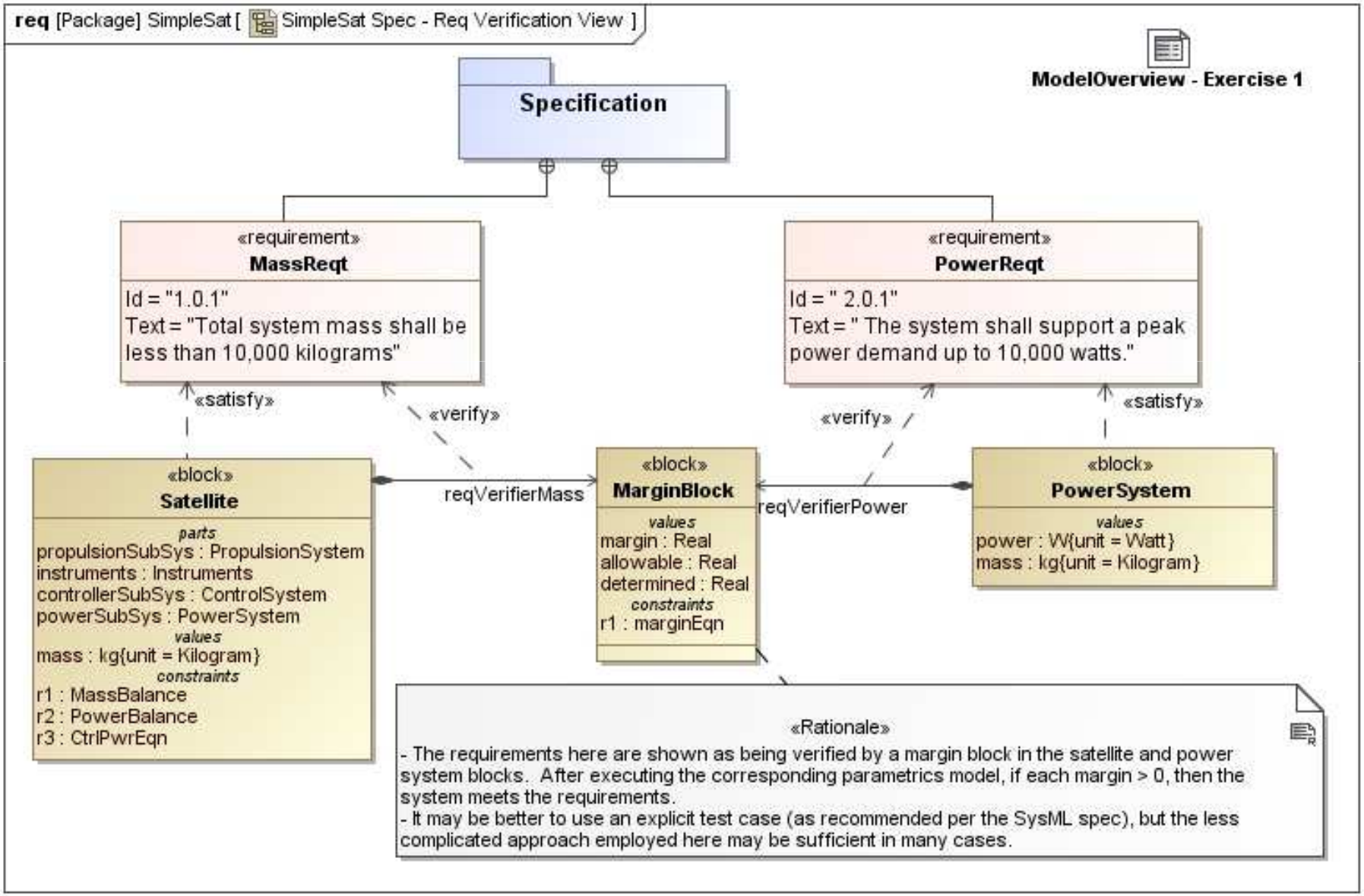
SimpleSat Parametrics Example

bdd depicting model structure



SimpleSat Parametrics Example

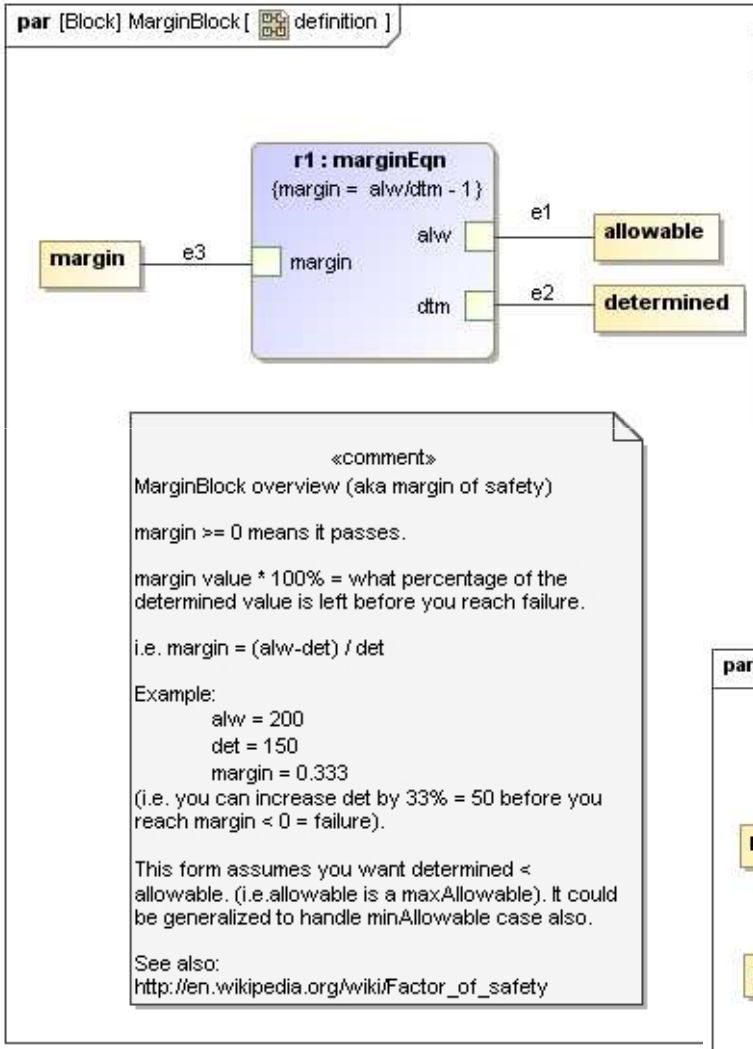
req diagram showing requirements verification pattern

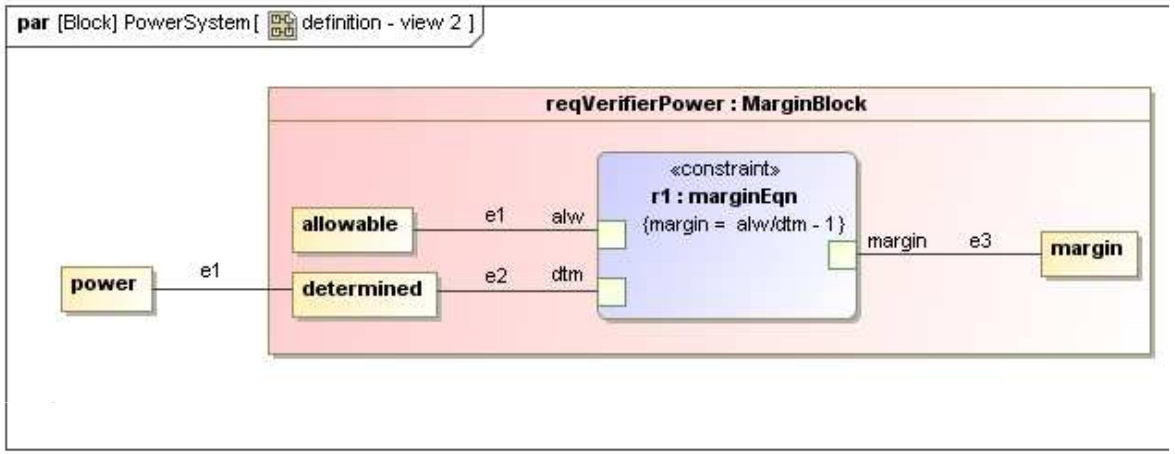


SimpleSat Parametrics Example

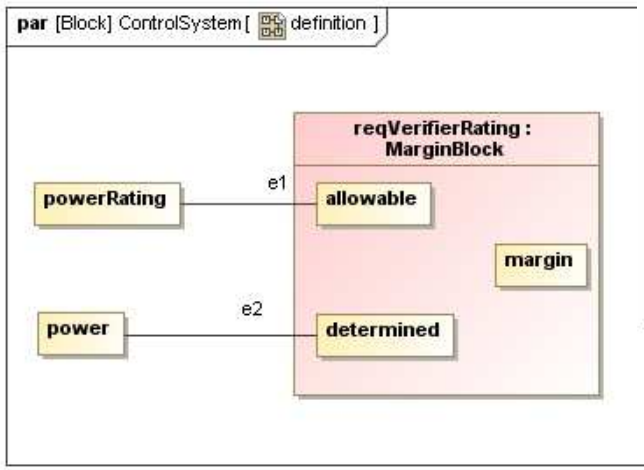
par structure of building blocks and subsystems

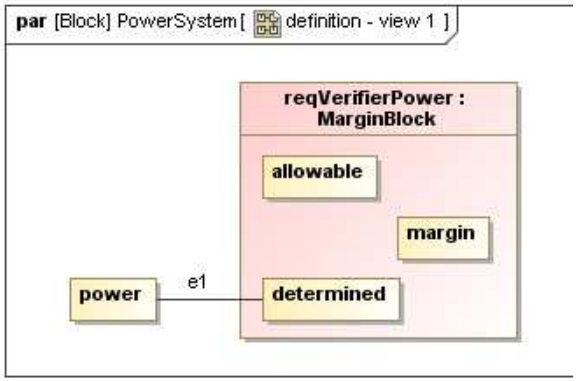
concept (generic): (a) expanded/flattened view





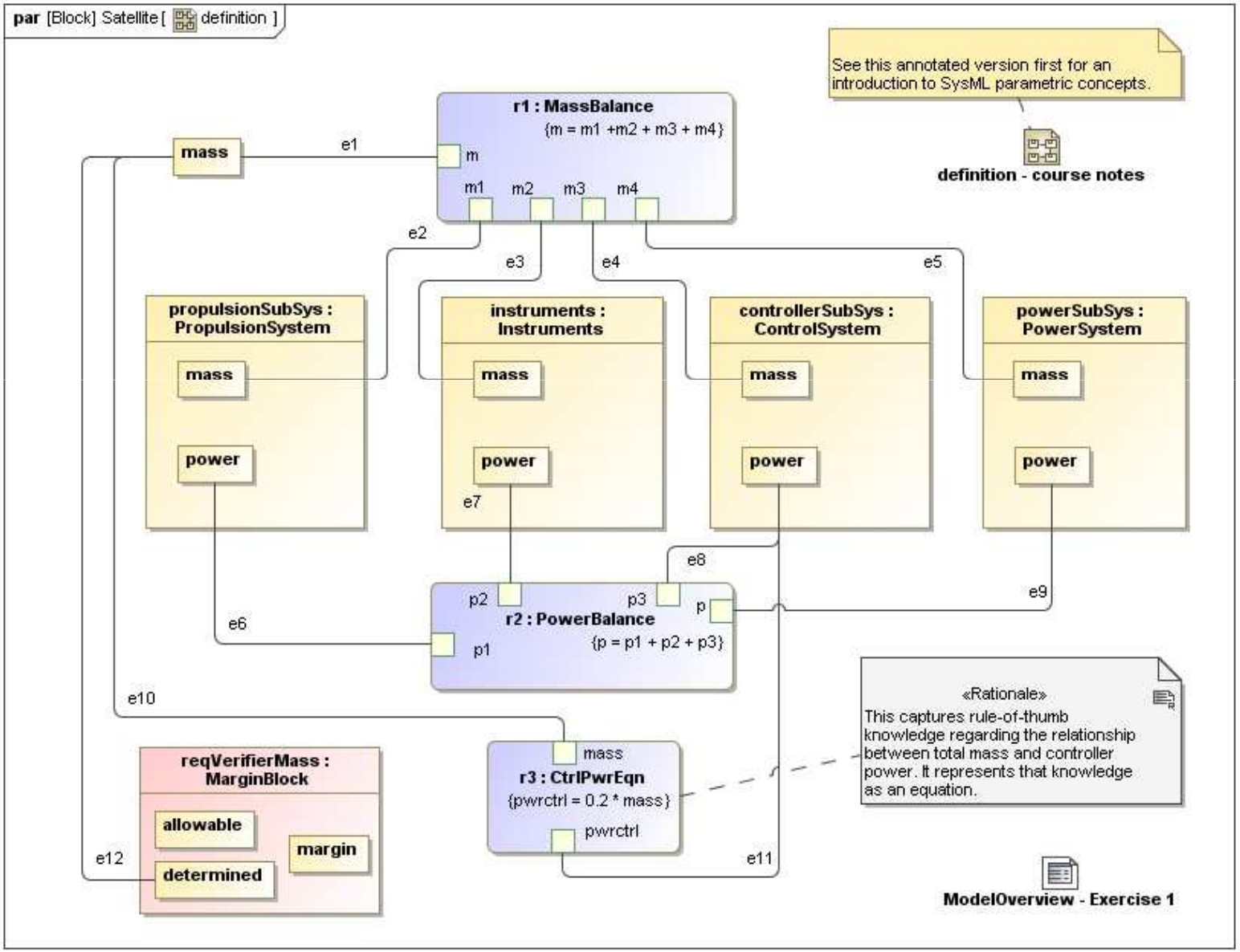
(b) encapsulated view





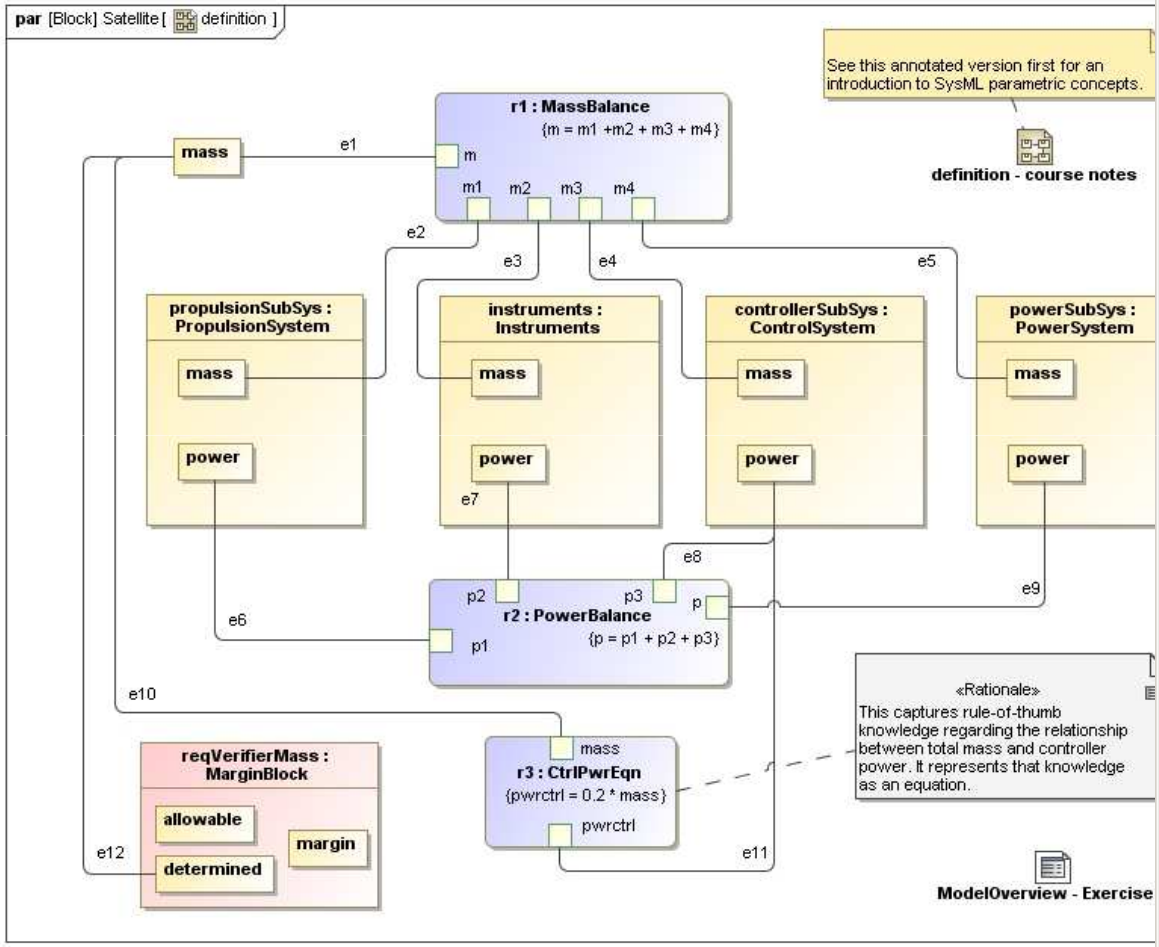
SimpleSat Parametrics Example

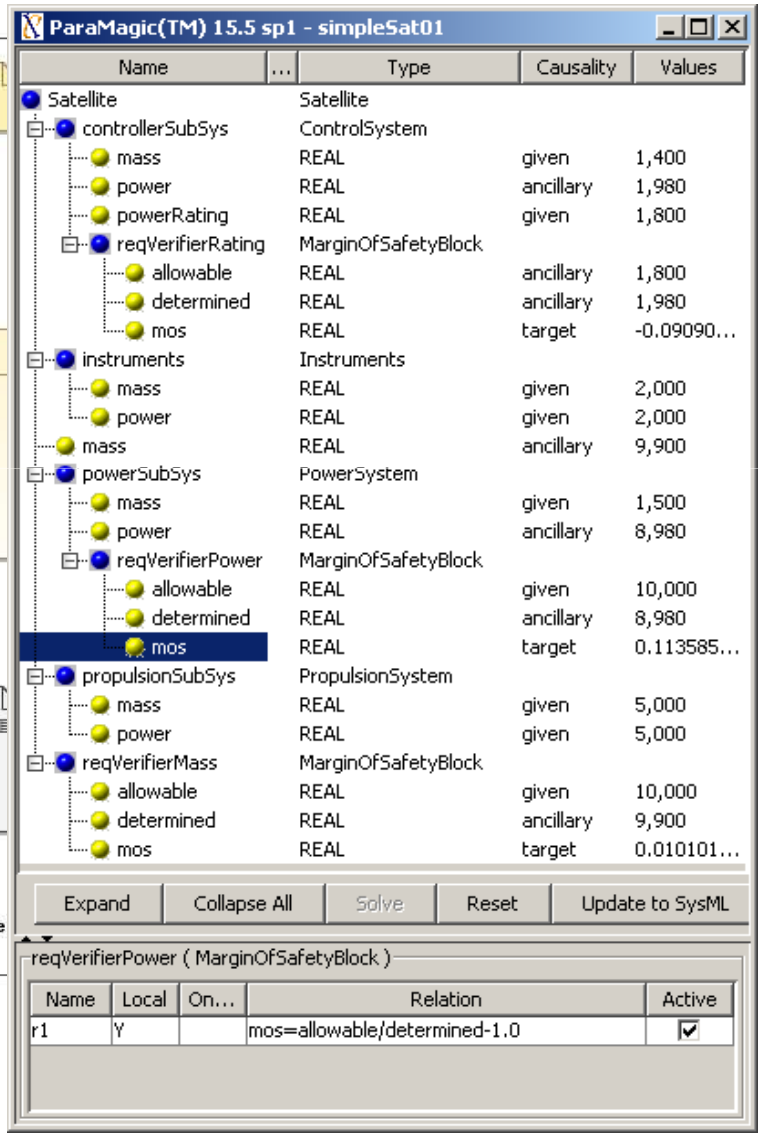
SysML parametrics diagram defining margin analysis



SimpleSat Parametrics Example

SysML par view and ParaMagic tool for execution

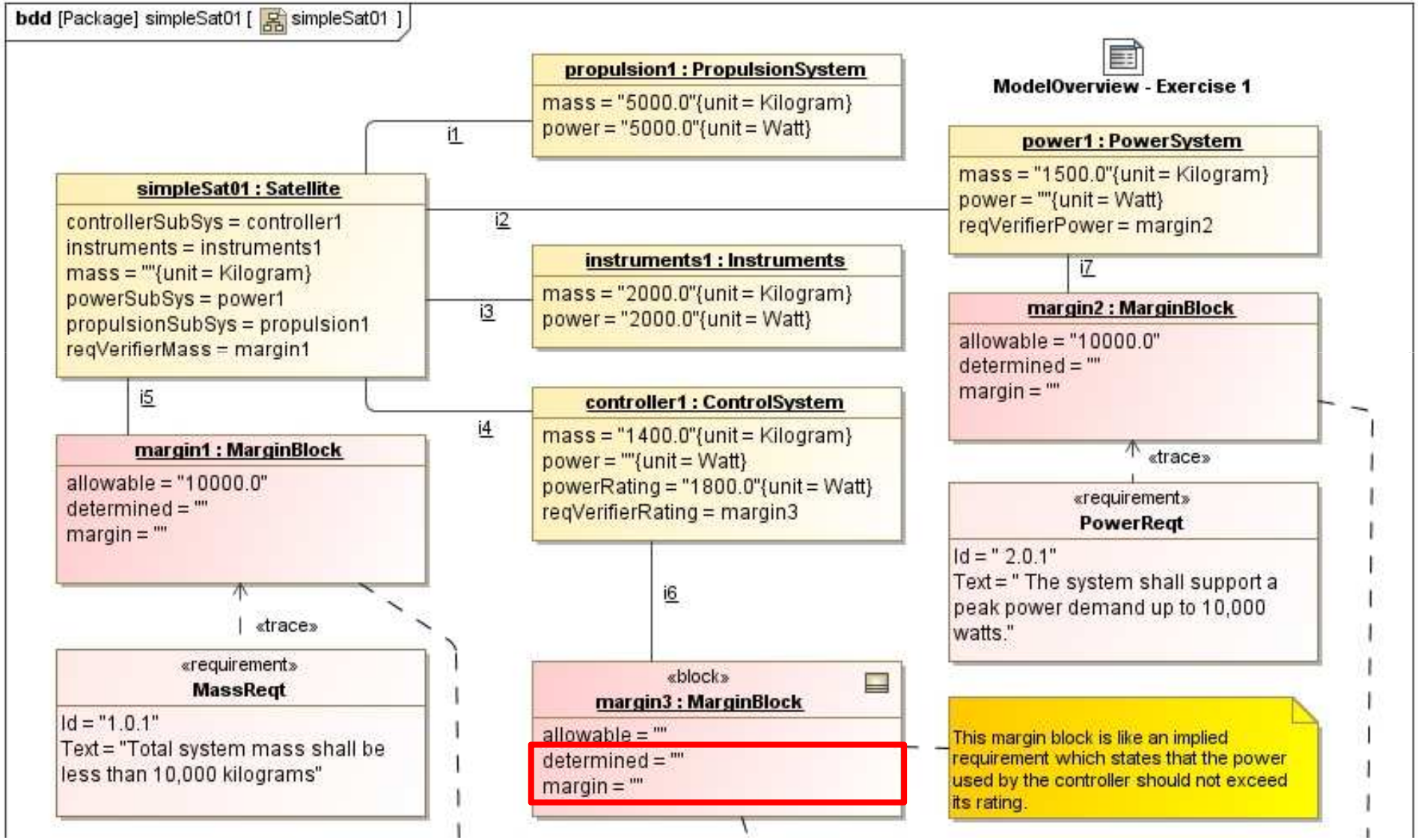




"Object-Oriented Spreadsheet++"

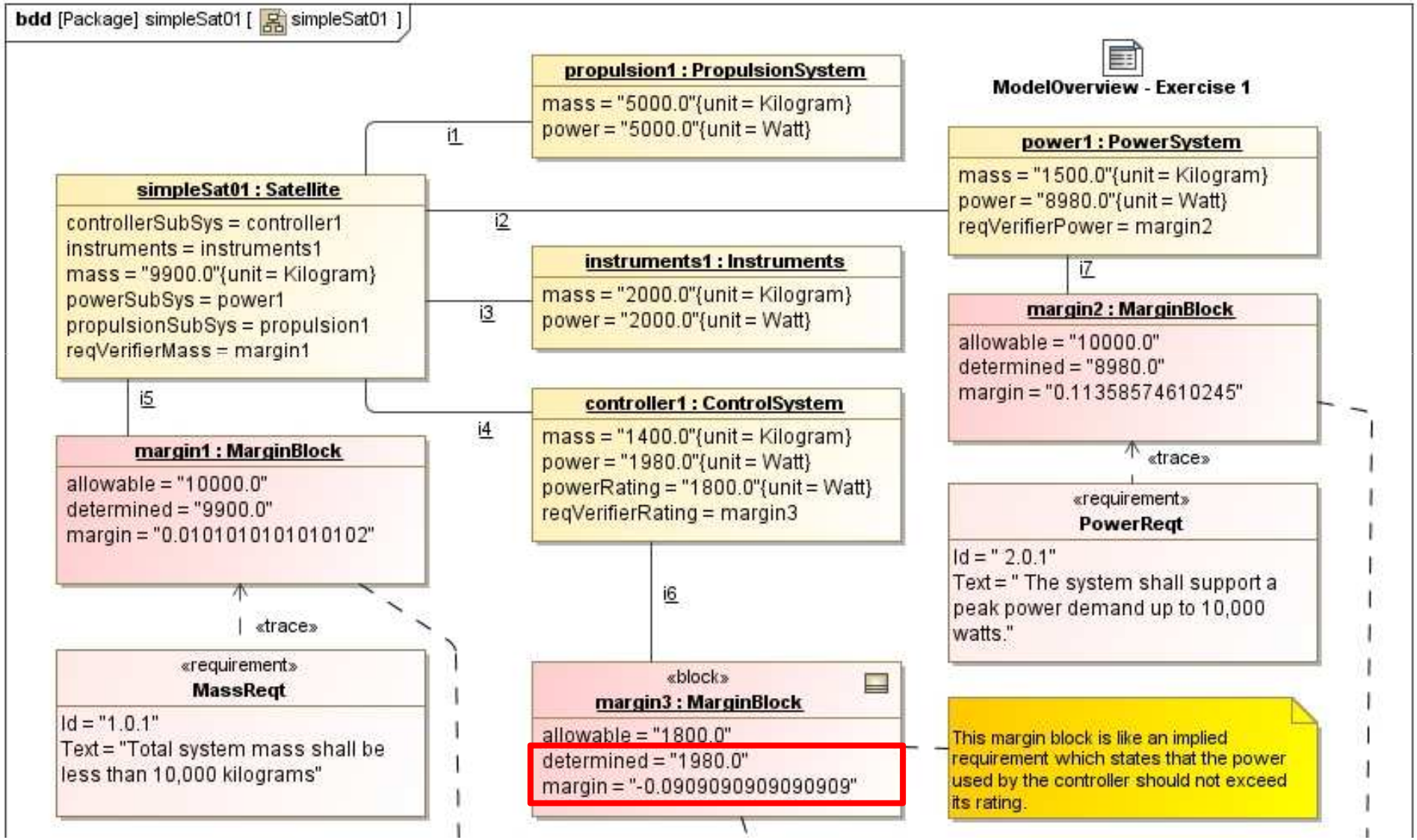
SimpleSat Parametrics Example

Analysis Results: a Satellite Instance



SimpleSat Parametrics Example

Analysis Results: a Satellite Instance



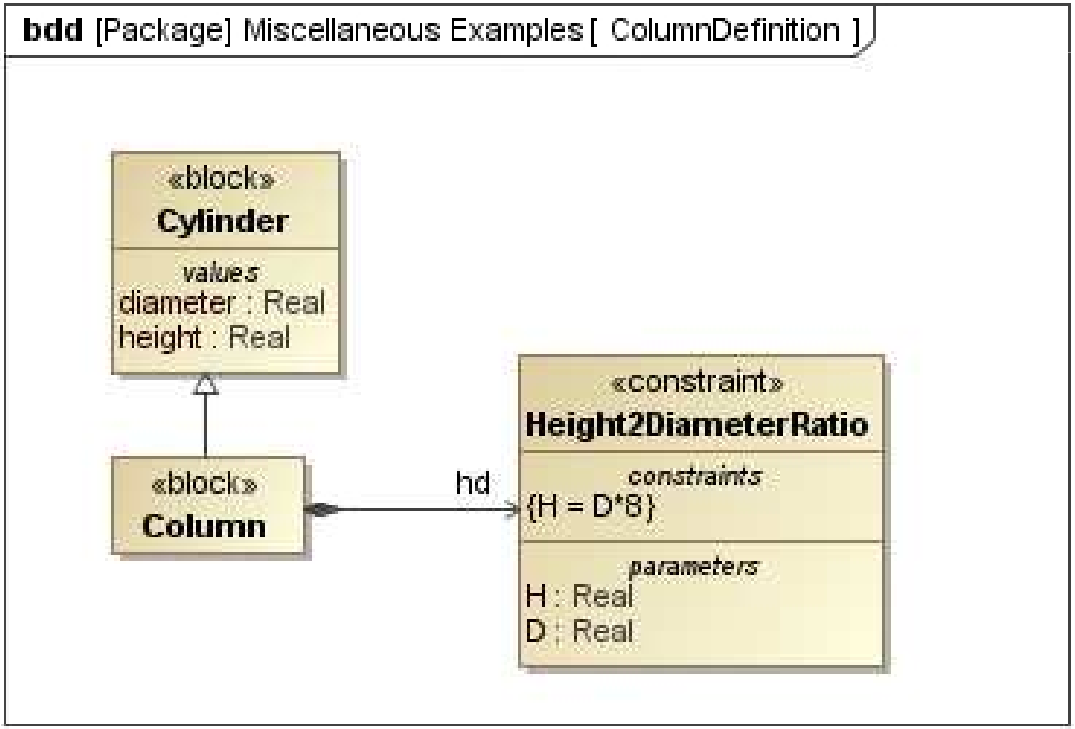
Outline

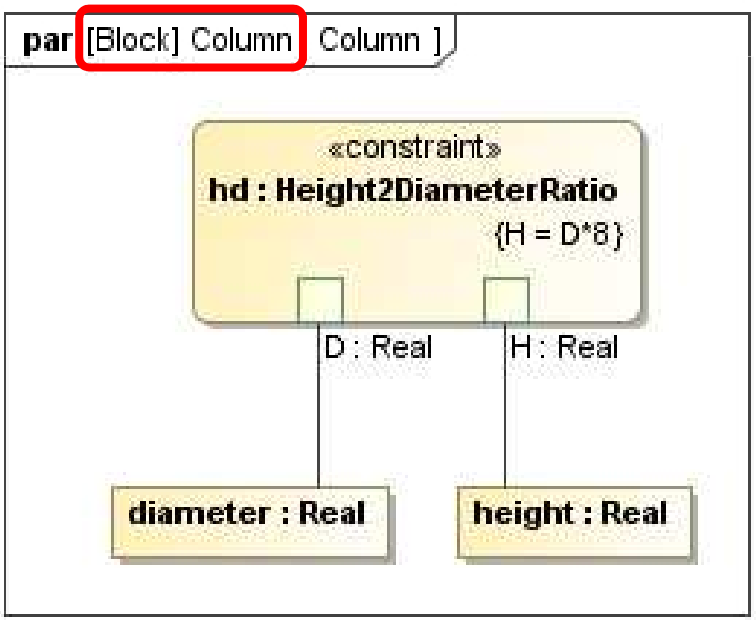
- Short overview of SysML
- SysML Parametrics: Syntax and Semantics
- Best practices in Paramatrics Modeling
  - Margin analysis – SimpleSat Satellite Example
  - Analysis context
  - Parameter aggregation
- Tools for solving Parametrics
  - ParaMagic
  - ModelCenter
- Parametrics versus SysML4Modelica
- Summary

Coffee Break

Two Different Uses of Parametrics: 1. Parametric Specification

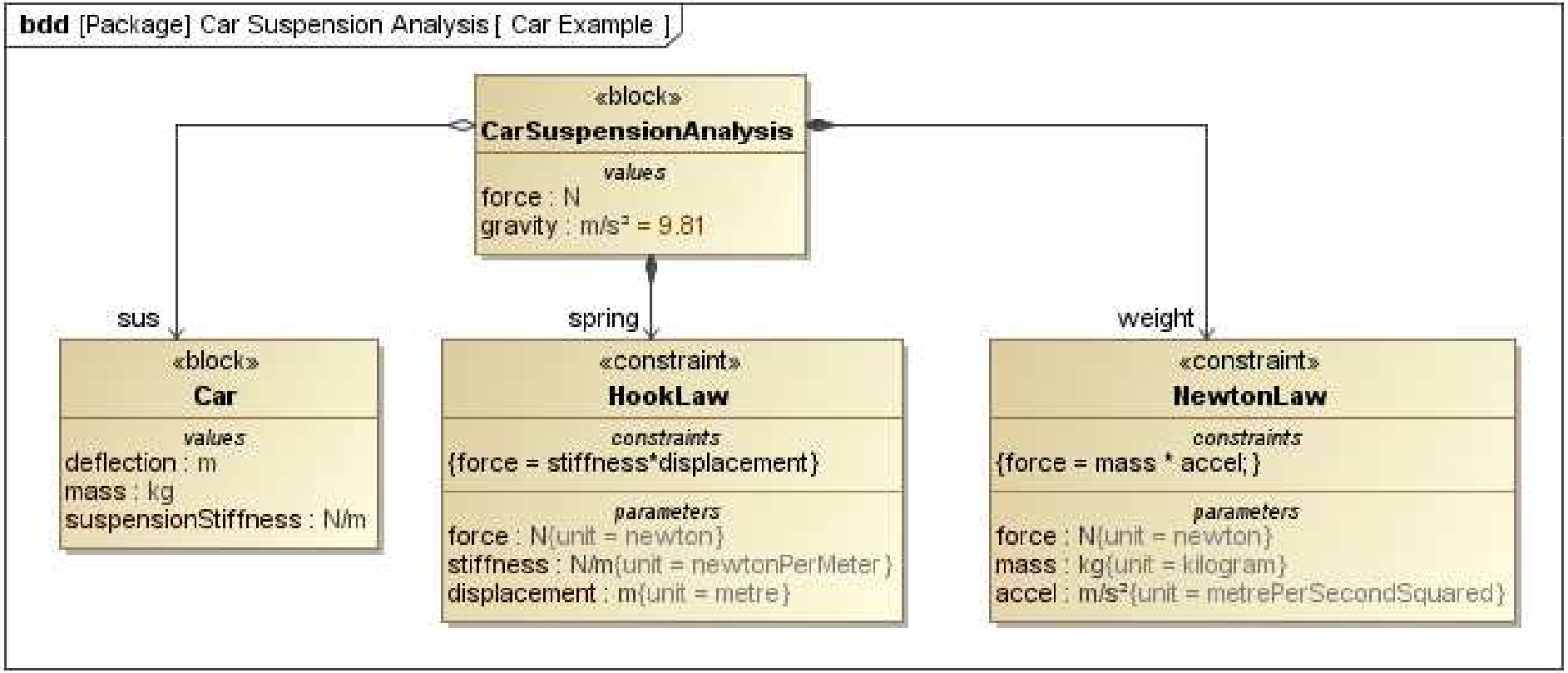
- A constraint is used to define a parametric relationship between value properties of a system alternative
- Constraint Property is owned by the Specified Block





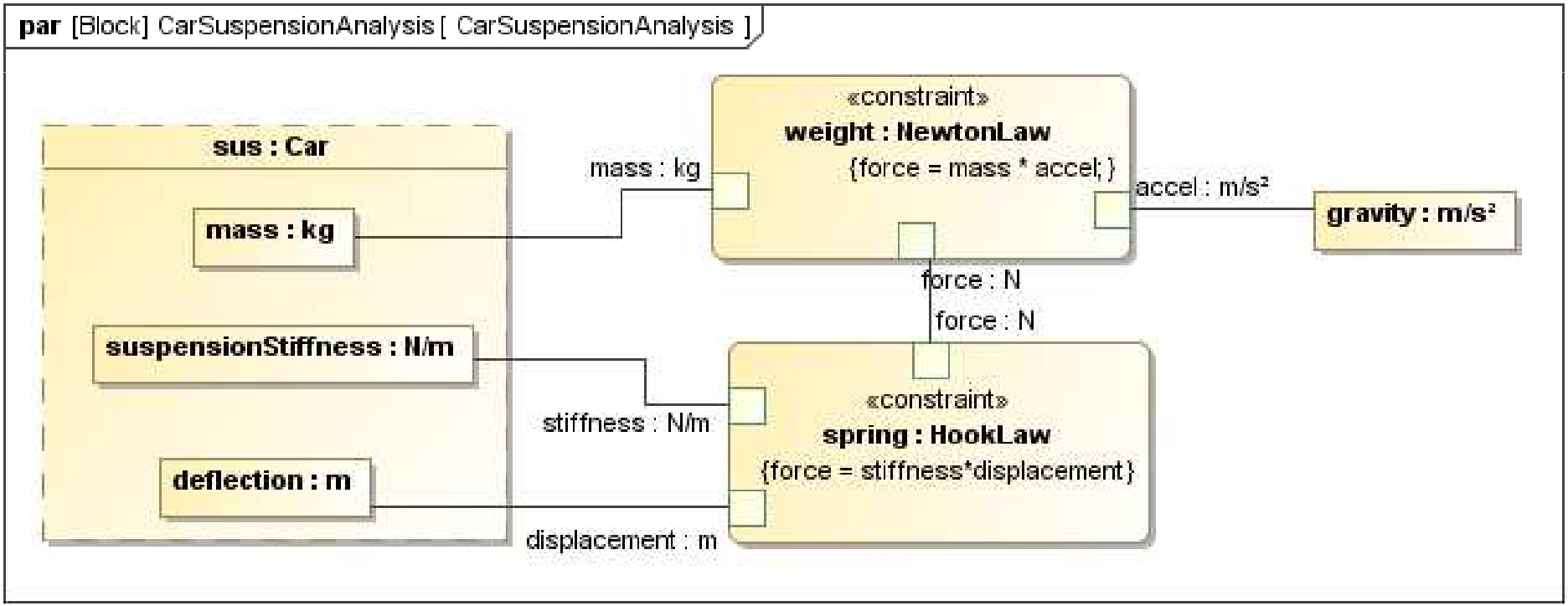
Two Different Uses of Parametrics: 2. Parametric Analysis

- A constraint is used to analyze an alternative
- Constraint Property is owned by an Analysis Context



Two Different Uses of Parametrics: 2. Parametric Analysis

- A constraint is used to analyze an alternative
- Different analyses are possible for the same system
- Constraint Property is owned by Analysis Context

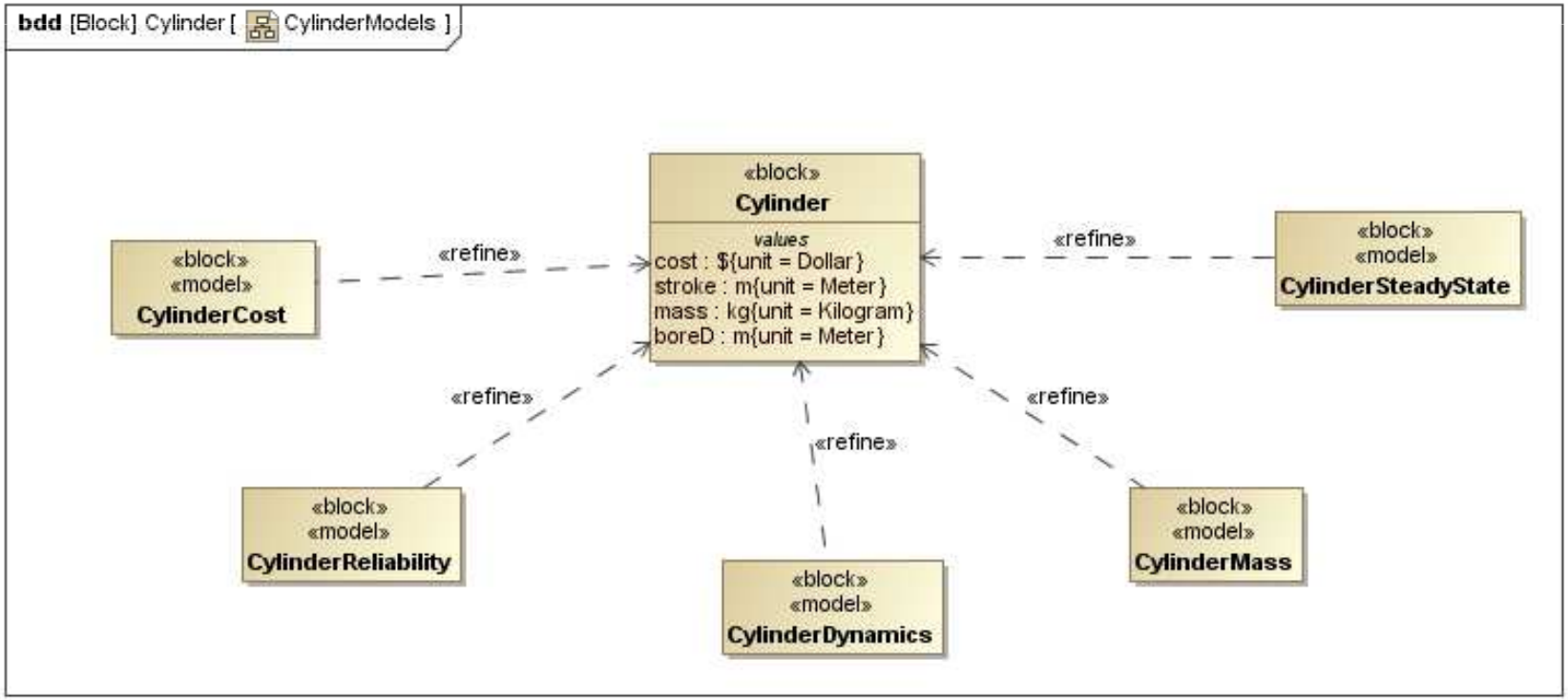


Why Use an Analysis Context?

- Many different analyses for a single system
  - Different views
  - Different levels of abstraction

One Context for each view / abstraction

- Analysis models often reused from libraries

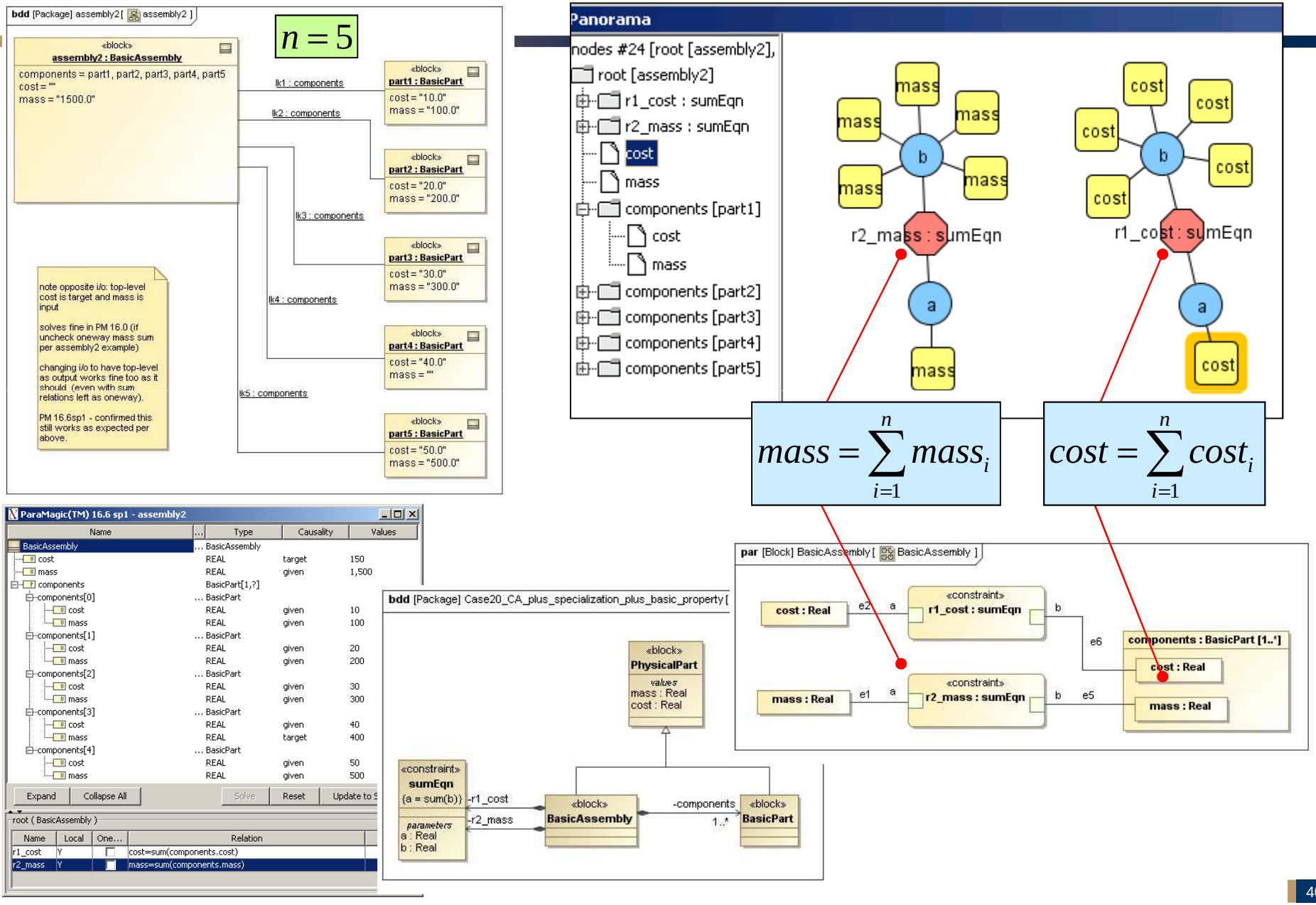


Outline

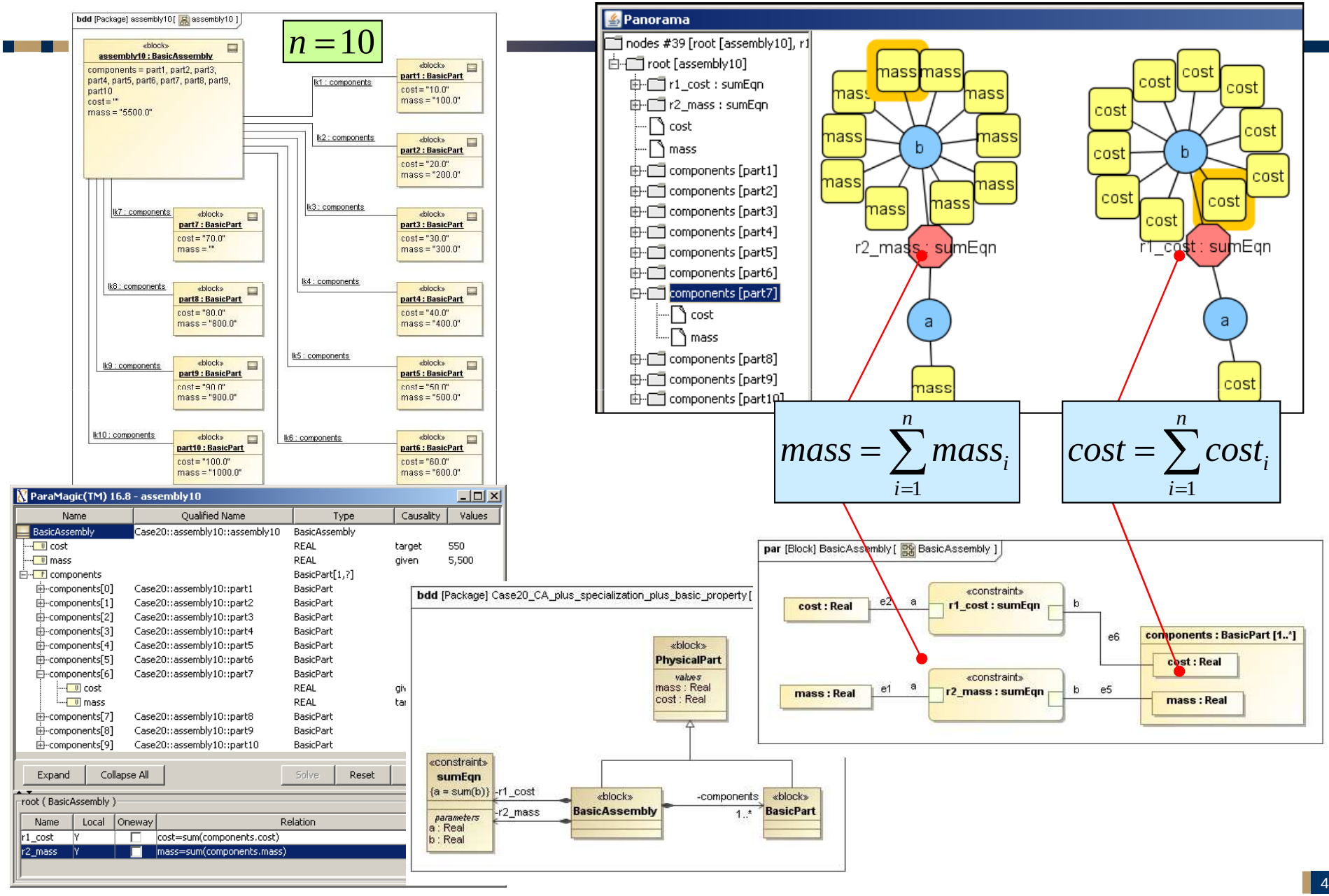
- Short overview of SysML
- SysML Parametrics: Syntax and Semantics
- Best practices in Paramatrics Modeling
  - Margin analysis – SimpleSat Satellite Example
  - Analysis context
  - Parameter aggregation
- Tools for solving Parametrics
  - ParaMagic
  - ModelCenter
- Parametrics versus SysML4Modelica
- Summary

Coffee Break

Complex Aggregates



Complex Aggregates



Outline

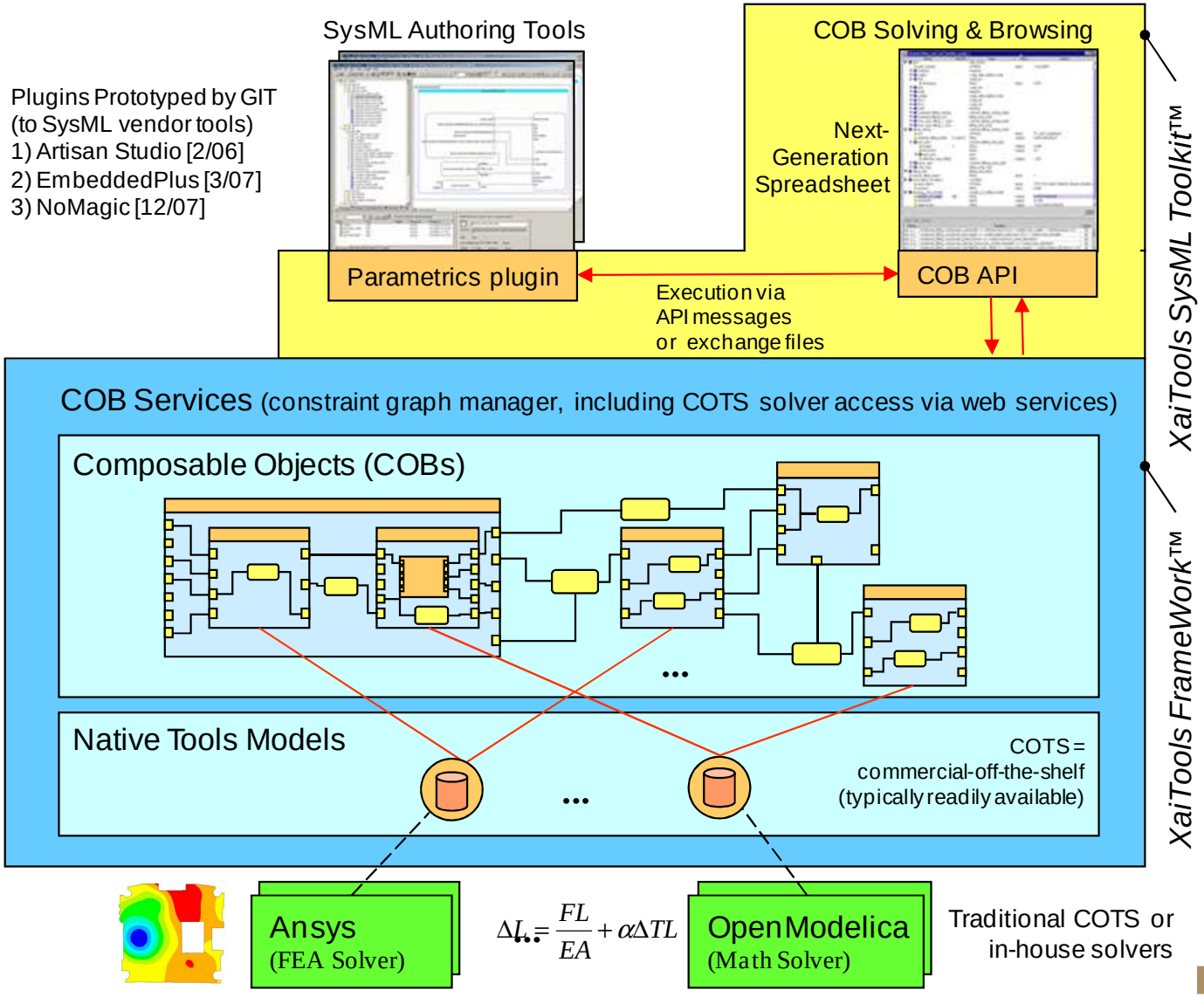
- Short overview of SysML
- SysML Parametrics: Syntax and Semantics
- Best practices in Paramatrics Modeling
  - Margin analysis – SimpleSat Satellite Example
  - Analysis context
  - Parameter aggregation
- Tools for solving Parametrics
  - ParaMagic
  - ModelCenter
- Parametrics versus SysML4Modelica
- Summary

Coffee Break

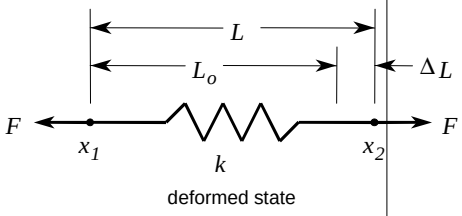
Enabling Executable SysML Parametrics

Commercialization by InterCAX LLC in Georgia Tech VentureLab incubator program

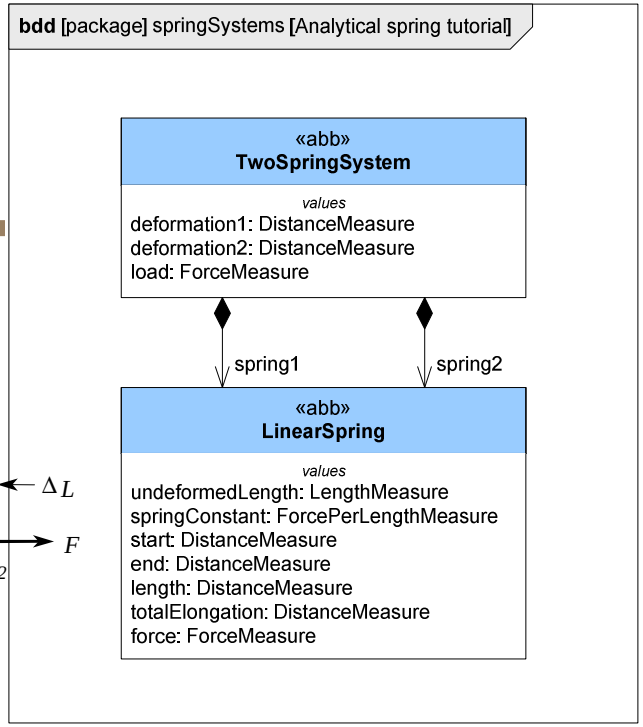
Advanced technology for graph management and solver access via web services.



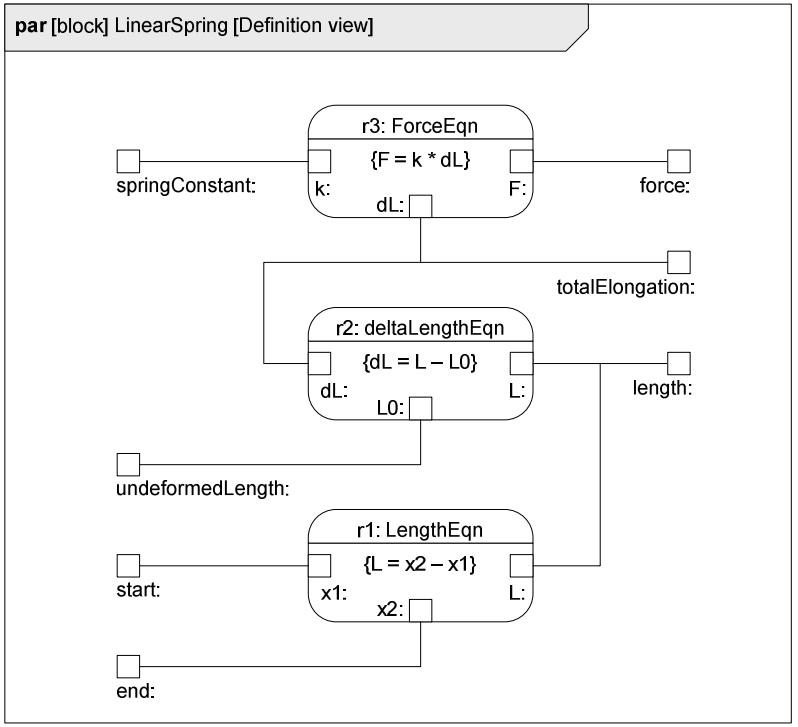
Spring System Example



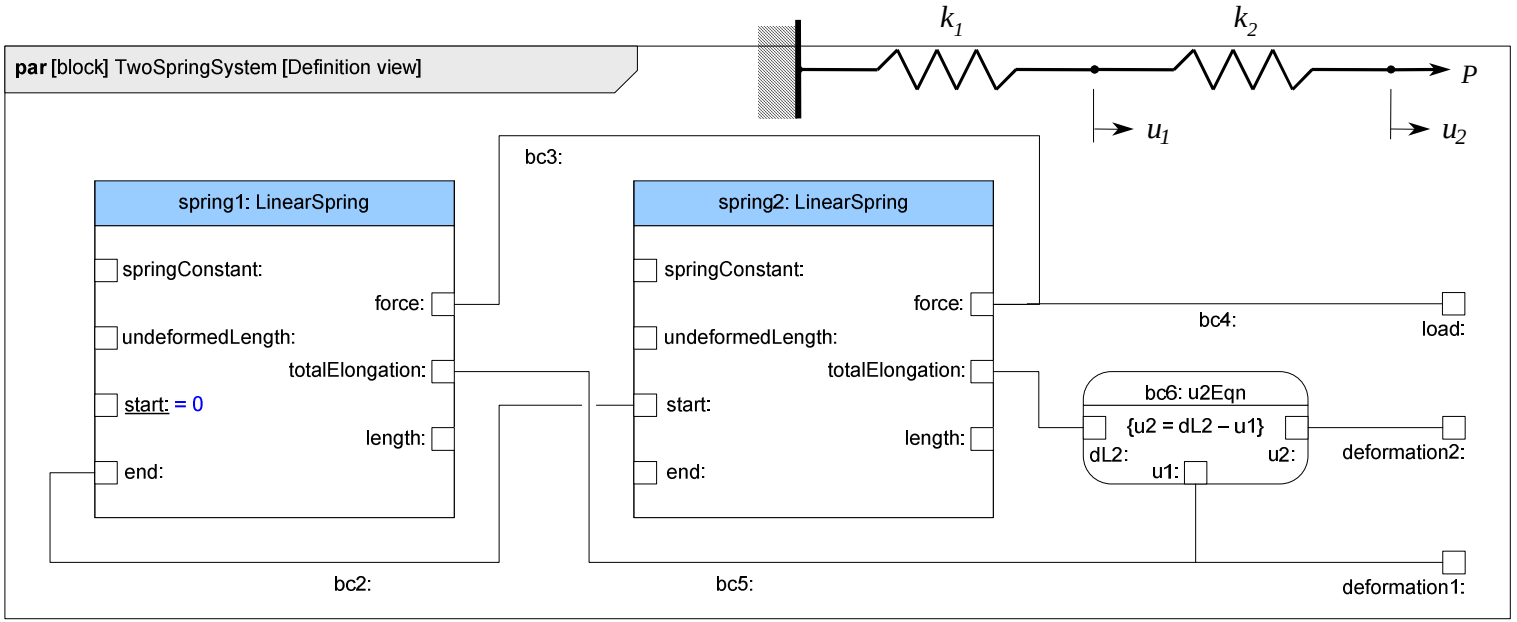
$$r_1 : L = x_2 - x_1$$
$$r_2 : \Delta L = L - L_0$$
$$r_3 : F = k\Delta L$$



(a) Analytical springs tutorial block definition diagram.



(b) LinearSpring parametric diagram.



(c) TwoSpringSystem parametric diagram.

Reference:
http://eislab.gatech.edu/pubs/conferences/2007-incose-is-1-peak-primer/

Example Instance: Two_Spring_System

two_spring_system

Name	Symbol	Type	Causality	Values
root		two_spring_system		
spring1		spring		
undeformed_length		REAL	given	8
spring_constant		REAL	given	5.5
start		REAL	undefined	?????
end0		REAL	undefined	?????
force		REAL	undefined	?????
total_elongation		REAL	undefined	?????
length0		REAL	undefined	?????
spring2		spring		
undeformed_length		REAL	given	8
spring_constant		REAL	given	6
start		REAL	undefined	?????
force		REAL	undefined	?????
total_elongation		REAL	undefined	?????
length0		REAL	undefined	?????
end0		REAL	undefined	?????
deformation1		REAL	target	?????
deformation2		REAL	target	?????
load		REAL	given	10

Solve local COBs

root (two_spring_system)

Name	Local	Oneway	Relation	Active
bc1	Y		<spring1.start> == 0.0	☑
bc2	Y		<spring1.end0> == <spring2.start>	☑
bc3	Y		<spring1.force> == <spring2.force>	☑
bc4	Y		<spring2.force> == <load>	☑
bc5	Y		<deformation1> == <spring1.total_elongation>	☑
bc6	Y		<deformation2> == <spring2.total_elongation> + <deformation1>	☑

example 2, state 1.0 (unsolved)

two_spring_system

Name	Symbol	Type	Causality	Values
root		two_spring_system		
spring1		spring		
undeformed_length		REAL	given	8
spring_constant		REAL	given	5.5
start		REAL	ancillary	0
end0		REAL	ancillary	9.818181818182
force		REAL	ancillary	10
total_elongation		REAL	ancillary	1.818181818182
length0		REAL	ancillary	9.818181818182
spring2		spring		
undeformed_length		REAL	given	8
spring_constant		REAL	given	6
start		REAL	ancillary	9.818181818182
force		REAL	ancillary	10
total_elongation		REAL	ancillary	1.666666666667
length0		REAL	ancillary	9.666666666667
end0		REAL	ancillary	19.484848484848
deformation1		REAL	target	1.818181818182
deformation2		REAL	target	3.484848484848
load		REAL	given	10

Solve local COBs

root (two_spring_system)

Name	Local	Oneway	Relation	Active
bc1	Y		<spring1.start> == 0.0	☑
bc2	Y		<spring1.end0> == <spring2.start>	☑
bc3	Y		<spring1.force> == <spring2.force>	☑
bc4	Y		<spring2.force> == <load>	☑
bc5	Y		<deformation1> == <spring1.total_elongation>	☑
bc6	Y		<deformation2> == <spring2.total_elongation> + <deformation1>	☑

example 2, state 1.1 (solved)

(b) Parametrics execution in *XaiTools / ParaMagic*

(a) Lexical COB instance as XML (CXI)

```
<linear_spring loid="_15">
  <undeformed_length causality="given">8.0</undeformed_length>
  <spring_constant causality="given">5.5</spring_constant>
</linear_spring>

<linear_spring loid="_25">
  <undeformed_length causality="given">8.0</undeformed_length>
  <spring_constant causality="given">6.0</spring_constant>
</linear_spring>
```

```
<two_spring_system loid="_3">
  <spring1 ref="_15"/>
  <spring2 ref="_25"/>
  <deformation1 causality="target"/>
  <deformation2 causality="target"/>
  <load causality="given">10.0</load>
</two_spring_system>
```

ParaMagic Core Solver: Mathematica

Mathematica Job — SpringSystems

(a) Input script
(auto-generated from ParaMagic)
example 2, state 1.0 (unsolved)

```
 ...
solutions = Solve[ {
  q16==k10,
  q16==o14*5.5,
  o14==n13,
  i8==j9-h7,
  10==k10,
  p15==g6-0,
  l11==m12+n13,
  g6==h7,
  k10==m12*6,
  m12==i8-8,
  o14==p15-8
} ];

WriteString[ output,
  ToString[ CForm [N [ solutions ]
] ] ];
Close[output];
Exit[];
```

(b) Output script (results)
(auto-imported back into ParaMagic)
example 2, state 1.1 (solved)

```
List(List(
      ...
  Rule(g6,9.818181818181818),
  Rule(h7,9.818181818181818),
  Rule(i8,9.666666666666666),
  Rule(j9,19.48484848484848),
  Rule(k10,10.),
  Rule(m12,1.6666666666666665),
  Rule(l11,3.484848484848485),
  Rule(n13,1.8181818181818183),
  Rule(o14,1.8181818181818183),
  Rule(p15,9.818181818181818),
  Rule(q16,10.)))      ...
))
```

Note: ParaMagic supports either of these as a core solver (in production releases): Mathematica and OpenModelica.
Support for Matlab Symbolic Math Toolbox (SMT) as a core solver is WIP.

ParaMagic Core Solver: OpenModelica

OpenModelica Job — SpringSystems

(a) Input script
(auto-generated from ParaMagic)

example 2, state 1.0 (unsolved)

```
class SpringSystems991034
   Real e4;    Real i8;
   Real l11;   Real a0;
   Real k10;   Real m12;
   Real b1;    Real d3;
   Real p15;   Real f5;
   Real o14;
equation
   10.0=l11;
   p15=m12-8.0;
   l11=p15*6.0;
   i8=f5-8.0;
   b1=p15+a0;
   m12=k10-o14;
   f5=d3-0.0;
   i8=a0;
   e4=l11;
   e4=i8*5.5;
   d3=o14;
end SpringSystems991034;
```

(b) Output script (results)
(auto-imported back into ParaMagic)

example 2, state 1.1 (solved)

```
         ...
DataSet: a0
0, 1.81818181818182
DataSet: k10
0, 19.48484848484849
DataSet: m12
0, 9.66666666666667
DataSet: b1
0, 3.48484848484849
DataSet: p15
0, 1.66666666666667
DataSet: o14
0, 9.81818181818182
DataSet: e4
0, 10
         ...
```

Note: ParaMagic supports either of these as a core solver (in production releases): Mathematica and OpenModelica.
Support for Matlab Symbolic Math Toolbox (SMT) as a core solver is WIP.

ParaMagic Capabilities

www.InterCAX.com

InterCAX

Modeling & Simulation Technology

Products

SysML Parametric Solvers

- Execute SysML parametric models using
  - ParaMagic® for MagicDraw
  - Melody™ for Rational Rhapsody
  - ParaSolver™ for Artisan Studio
- Use Mathematica or OpenModelica (free) as a core solver
- Link to Excel spreadsheets and perform batch trade studies
- Link to existing MATLAB/Simulink and Mathematica functions

SysML Integration Tools

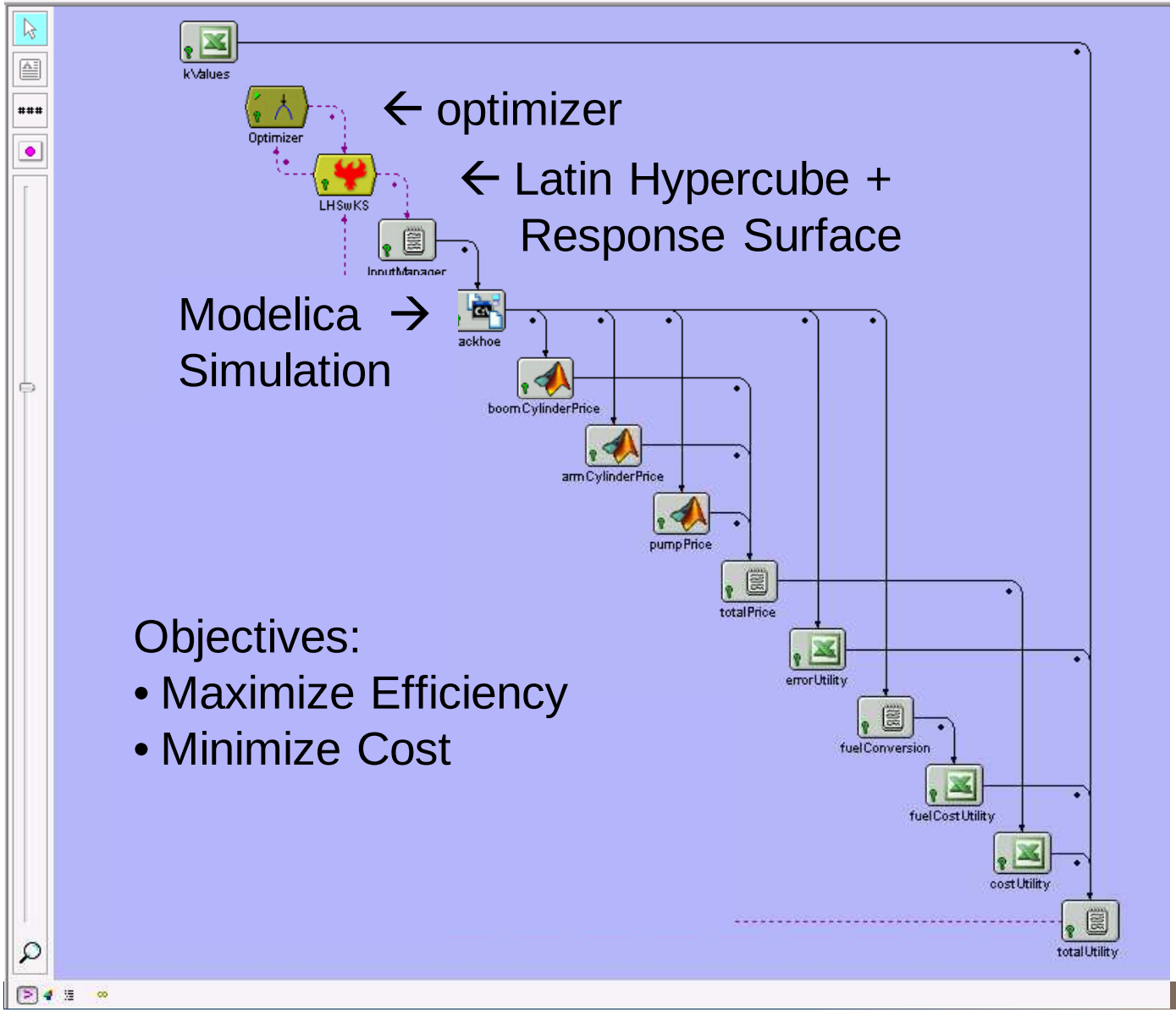
- SysML and Math tools (Mathematica, MATLAB, Excel)
- SysML and M&S tools (Simulink, STK, OpenModelica)
- SysML and CAD tools (NX, STEP MCAD/AP203, ECAD/AP210)
- SysML and CAE tools (ANSYS, ABAQUS)
- Plus tailored interfaces (Teamcenter/PLM, in-house tools)

Outline

- Short overview of SysML
- SysML Parametrics: Syntax and Semantics
- Best practices in Paramatrics Modeling
  - Margin analysis – SimpleSat Satellite Example
  - Analysis context
  - Parameter aggregation
- Tools for solving Parametrics
  - ParaMagic
  - ModelCenter
- Parametrics versus SysML4Modelica
- Summary

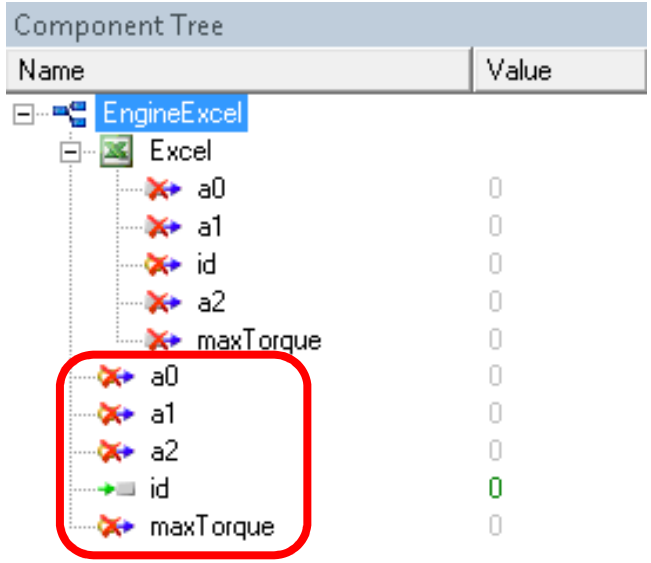
Coffee Break

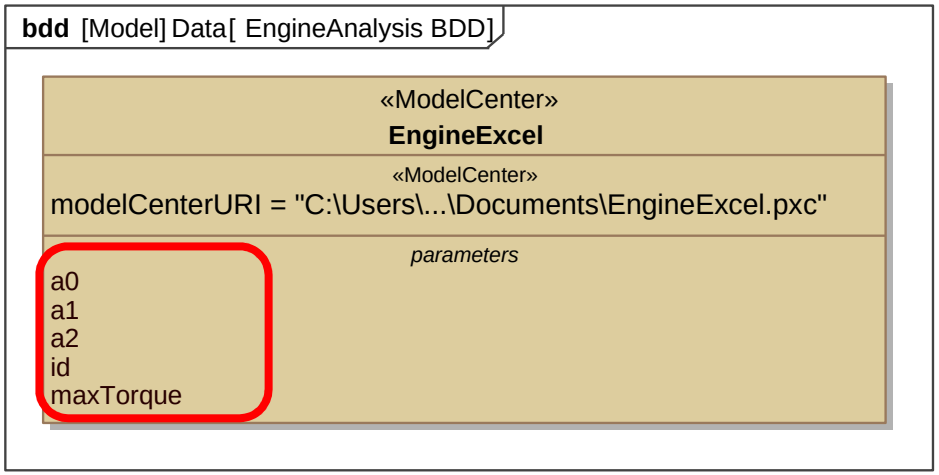
What is ModelCenter?



- Process Integration and Design Optimization (PIDO) framework
- Integration with most Engineering Analysis tools
- Supports grid computing

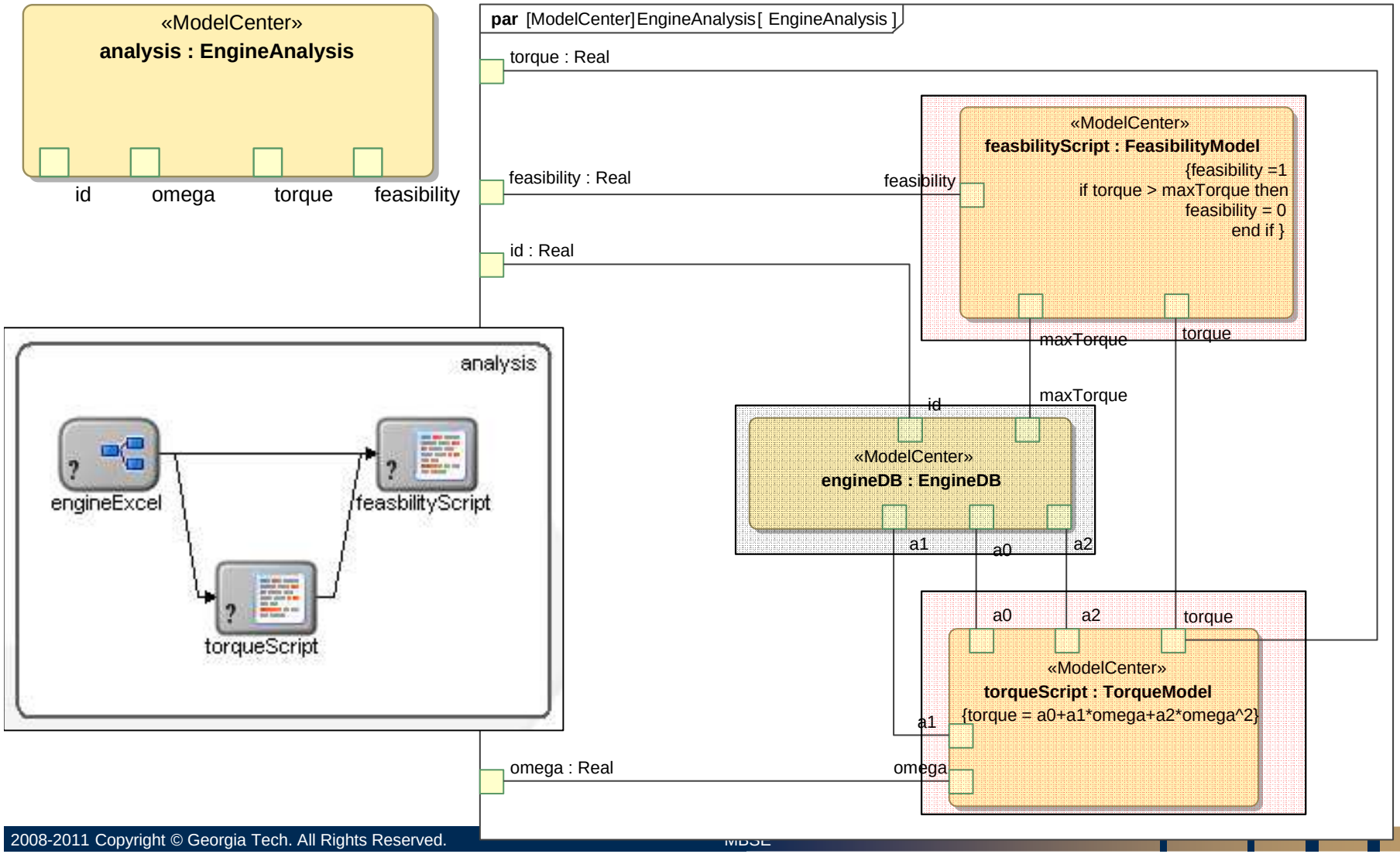
ModelCenter Model as Constraint Block





- Constraint block stereotyped to «ModelCenter»
- ModelCenter variables = SysML constraint parameters
- MagicDraw plug-in supports import-export
- Additional access to optimization, DOE, UQ, etc.

Nested Constraints in ModelCenter



ModelCenter Advantage / Disadvantages

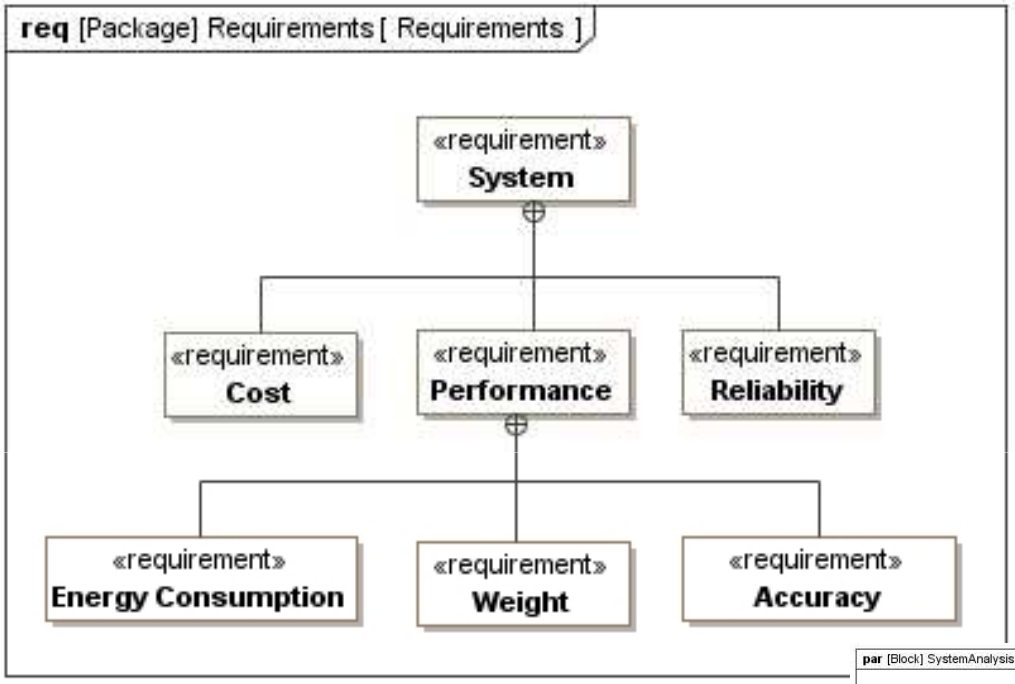
- Pro
  - One simple and convenient interface from MagicDraw to/from ModelCenter
  - Access to all the tools supported by ModelCenter
    - Math: Matlab, Excel,
    - Analysis: many CAE and CFD tools, Simulation tools,
    - Optimization: Dakotah, Design Explorer, …
    - CAD tools: CATIA, Pro/ENGINEER, NX
    - Specialized tools: STK, SEER, easy custom wrappers
  - Scalable: model management, grid computing, etc.
- Con
  - More expensive
  - Only causal models

Outline

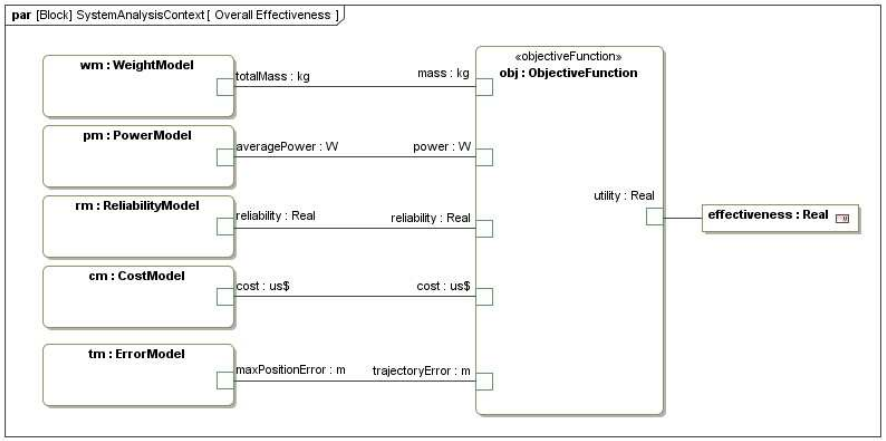
- Short overview of SysML
- SysML Parametrics: Syntax and Semantics
- Best practices in Paramatrics Modeling
  - Margin analysis – SimpleSat Satellite Example
  - Analysis context
  - Parameter aggregation
- Tools for solving Parametrics
  - ParaMagic
  - ModelCenter
- Parametrics versus SysML4Modelica
- Summary

Coffee Break

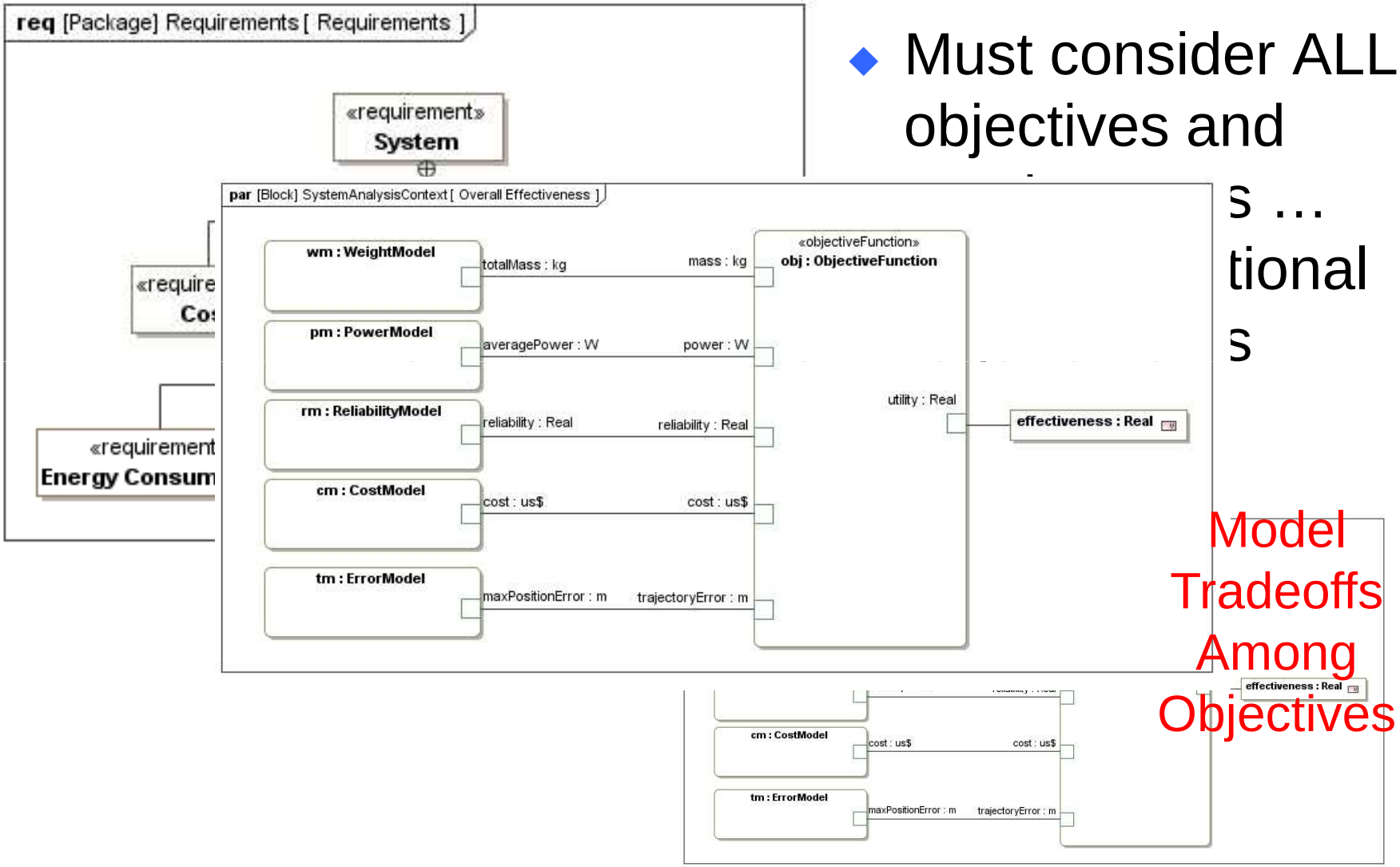
Modeling System-Level Trade-offs



- Must consider ALL objectives and requirements … not just functional requirements



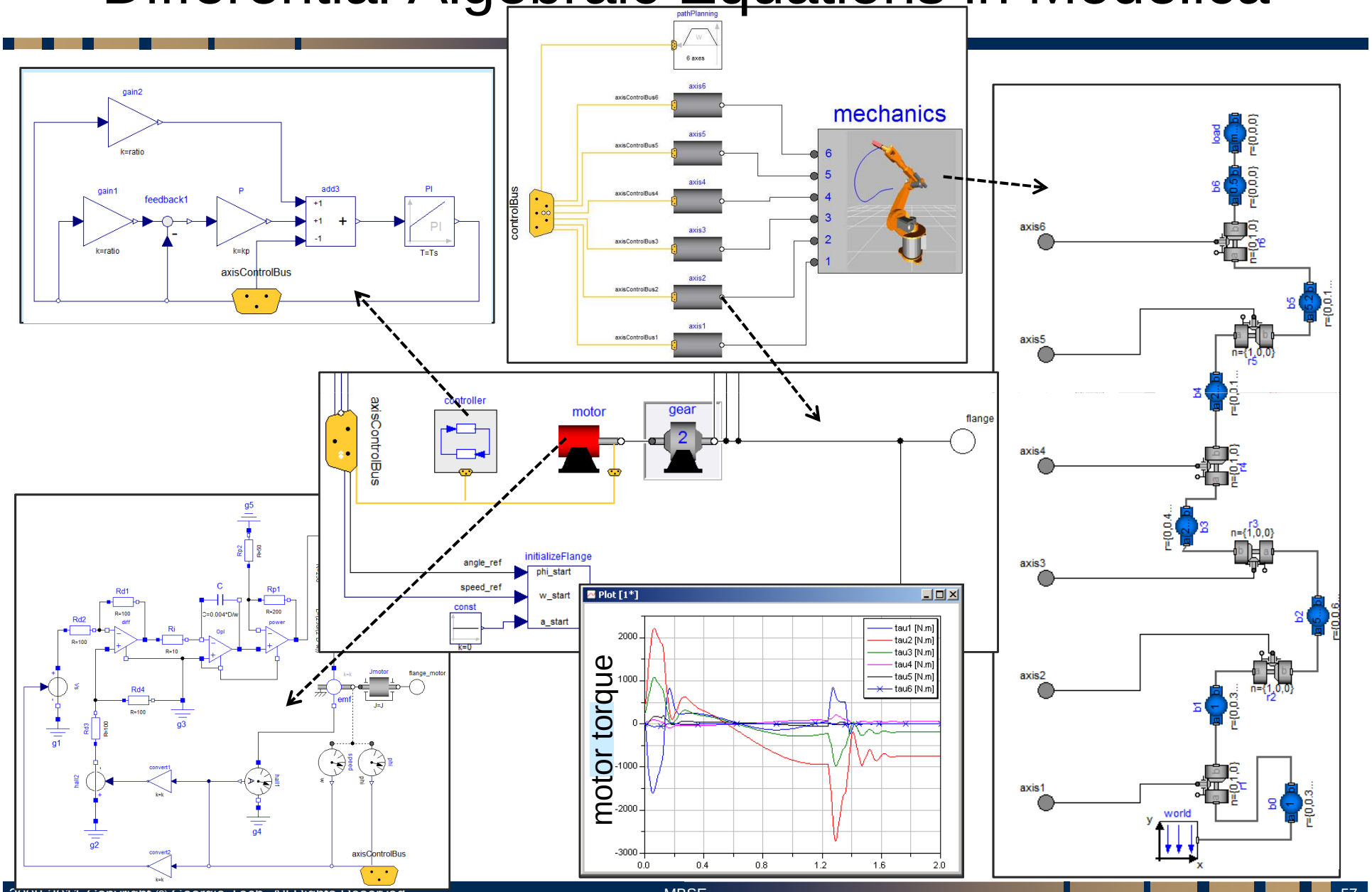
Modeling System-Level Trade-offs



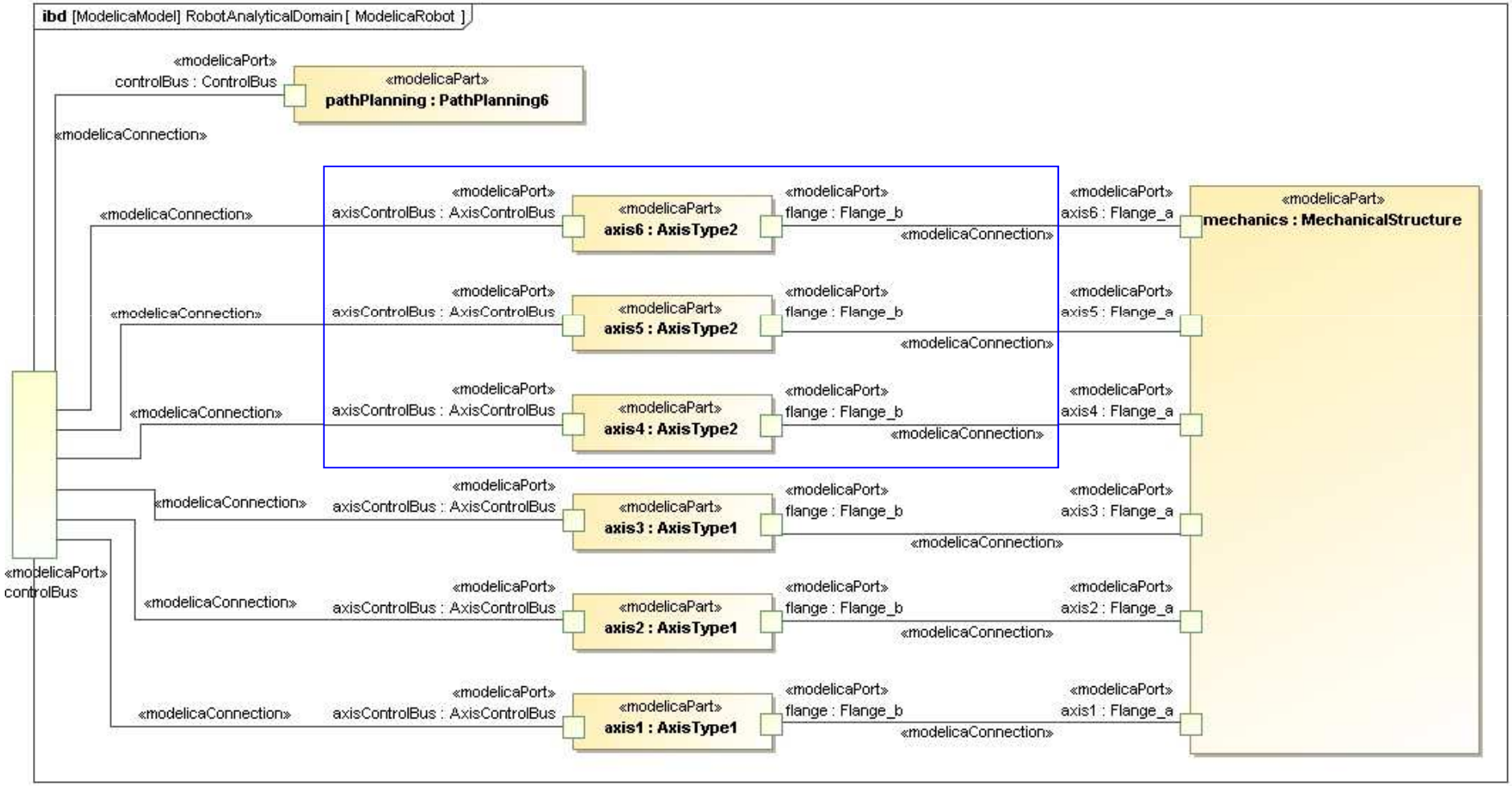
- Must consider ALL objectives and
  s …
  tional
  s

Model Tradeoffs Among Objectives

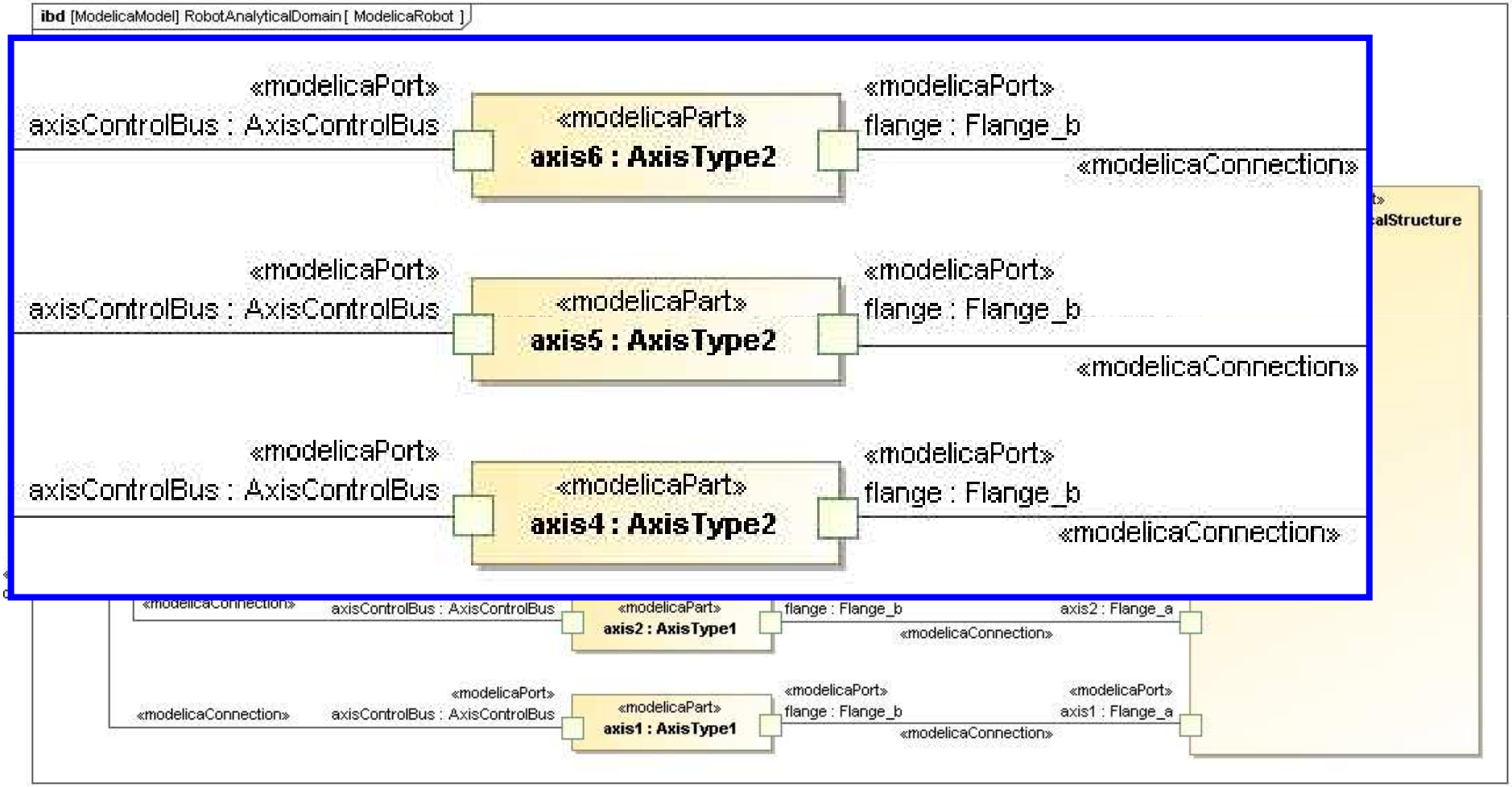
Modeling the Physics of Systems — Differential Algebraic Equations in Modelica



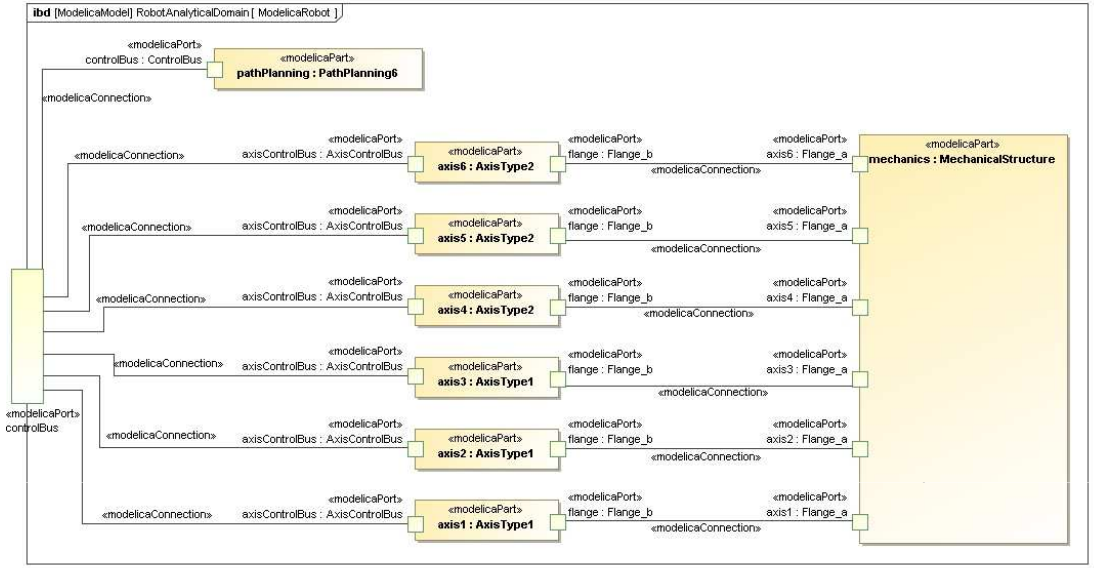
OMG SysML-Modelica Transformation Specification



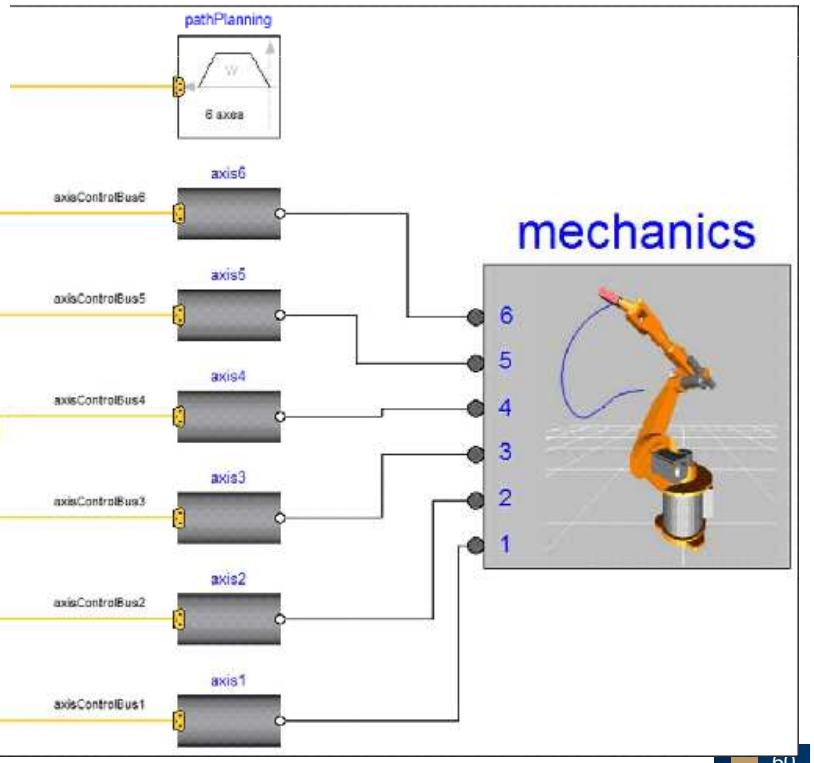
OMG SysML-Modelica Transformation Specification

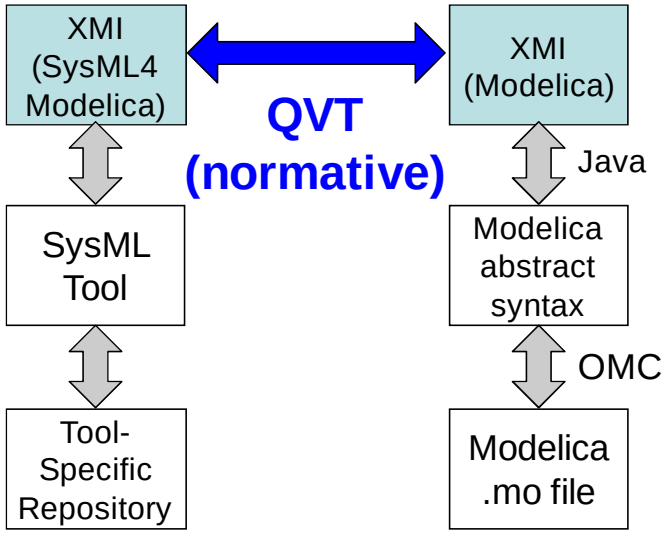


OMG SysML-Modelica Transformation Specification



Automatic Transformation





SysML Parametrics Discussion ...

- Advantages — disadvantages?
- Compared with Excel?
- Compared with Modelica?

Summary

- Short overview of SysML
- SysML Parametrics: Syntax and Semantics
- Best practices in Paramatrics Modeling
  - Margin analysis – SimpleSat Satellite Example
  - Analysis context
  - Model reuseParameter aggregation
- Tools for solving Parametrics
  - ParaMagic
  - ModelCenter
- Parametrics versus SysML4Modelica
- Summary