Extending SysML for Integration with Solver-based Simulation Tools

Ion Matei

Conrad Bock

National Institute of Standards and Technology

Overview

§ **Motivation and approach**

§ **Dynamic simulation overview**

§ **SysML extension**

§ **Detailed example**

§ **Transforming to simulation formats**

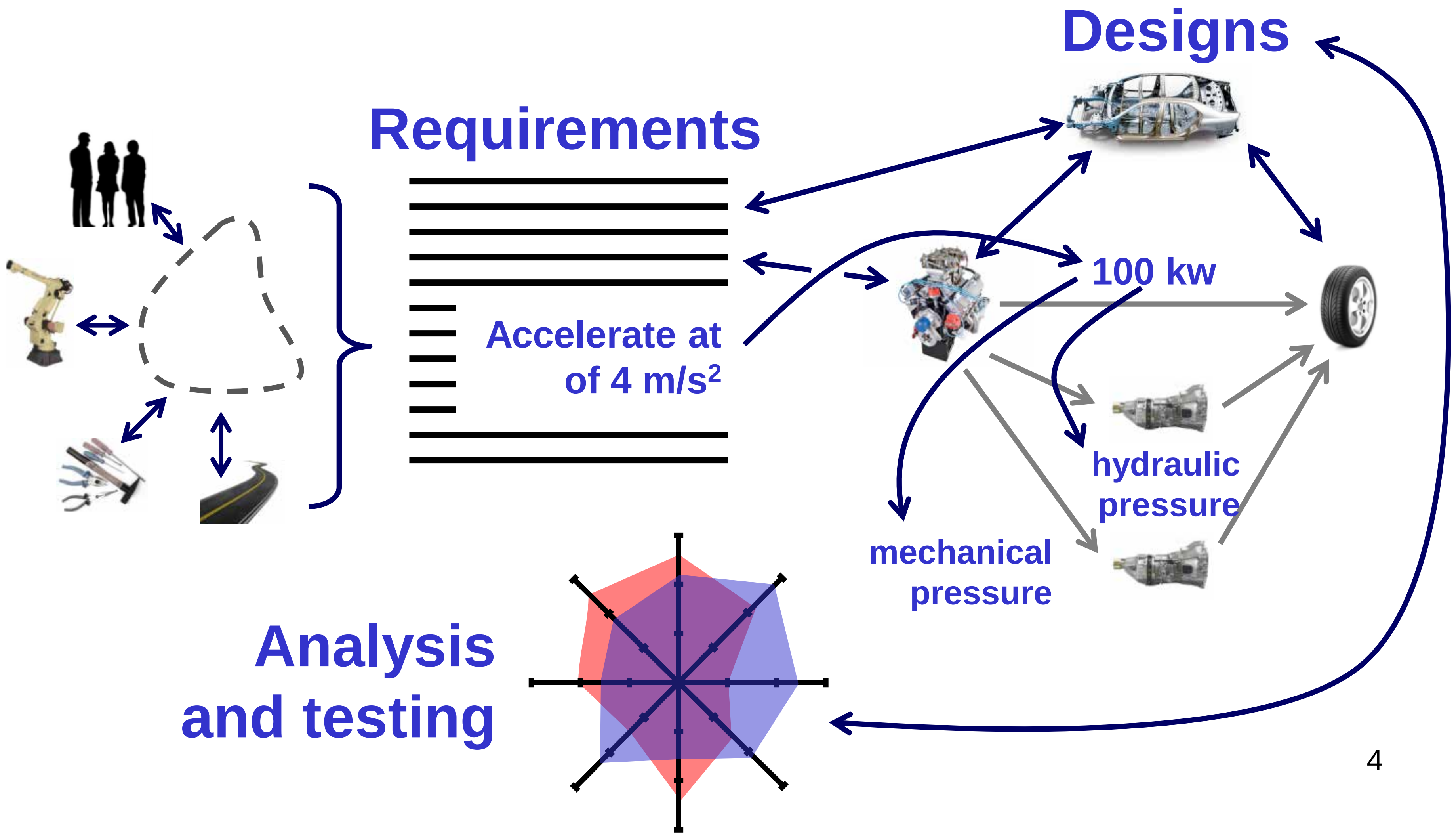
§ **Summary**

Overview

- **Motivation and approach**
- Dynamic simulation overview
- SysML extension
- Detailed example
- Transforming to simulation formats
- Summary

Model-based Systems Engineering

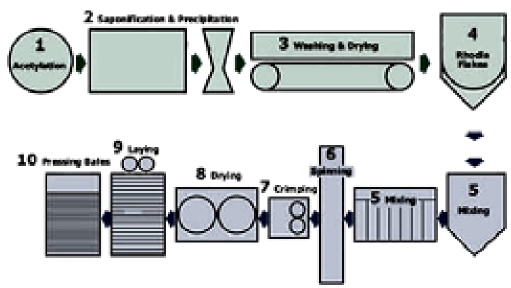
- Enabled by integrated models of:



Modeling Languages

- Needed for people / computers to share models.

Diagrams and/or text



1. Acetylation
2. Saponification & Precipitation
3. Washing & Drying
4. Rhodia Flakes
5. Mixing
6. Spinning
7. Crimping
8. Drying
9. Laying
10. Pressing Bales

- **Graphics:**
  - Circles,
  - Rectangles
  - Lines
- **Text:**
  - Reserved words

- **Domain terms:**
  - Lathes, Feeders
  - Drying, Shaping

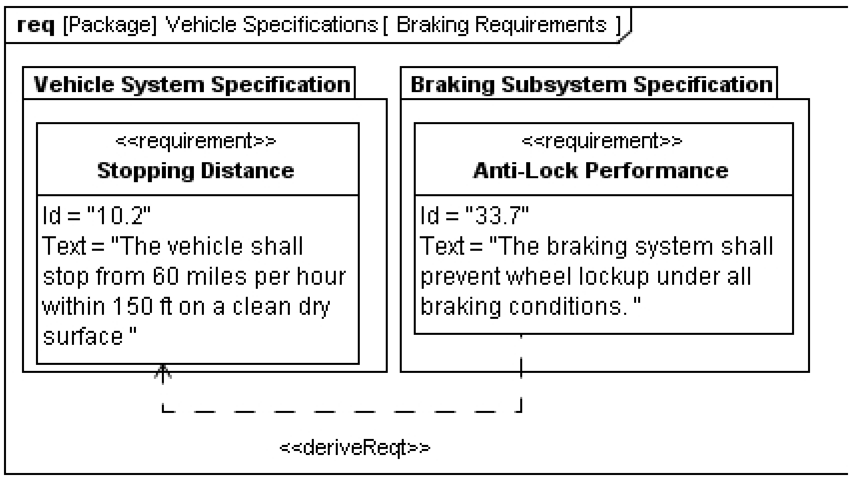
- **What happens:**
  - Geometry changed.
  - Pieces mounted onto machines.
  - Water removed.

Systems Modeling Language (SysML)

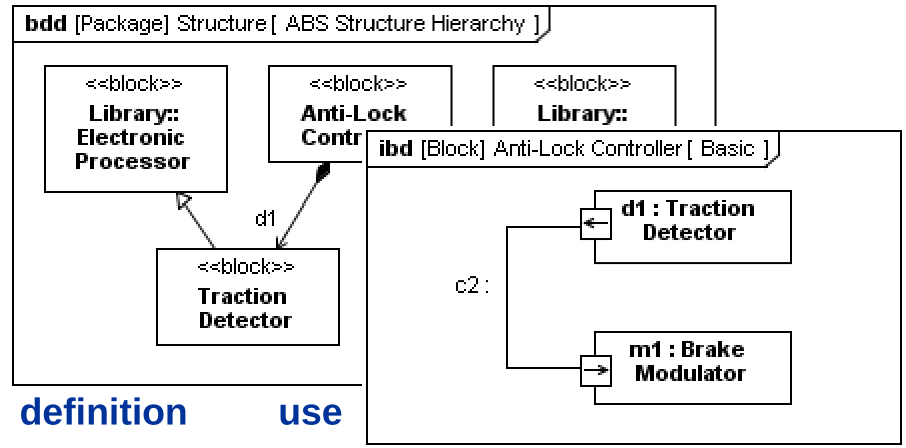
- Most widely used graphical modeling language for systems engineering.
- Open standard published by the Object Management Group (OMG).
- Initiated by the International Council on Systems Engineering (INCOSE).
- First published in 2007, most recent update in 2012.
- Adopted by practically all commerical and open source SE modeling tools.

SysML Diagrams

Requirements



Components

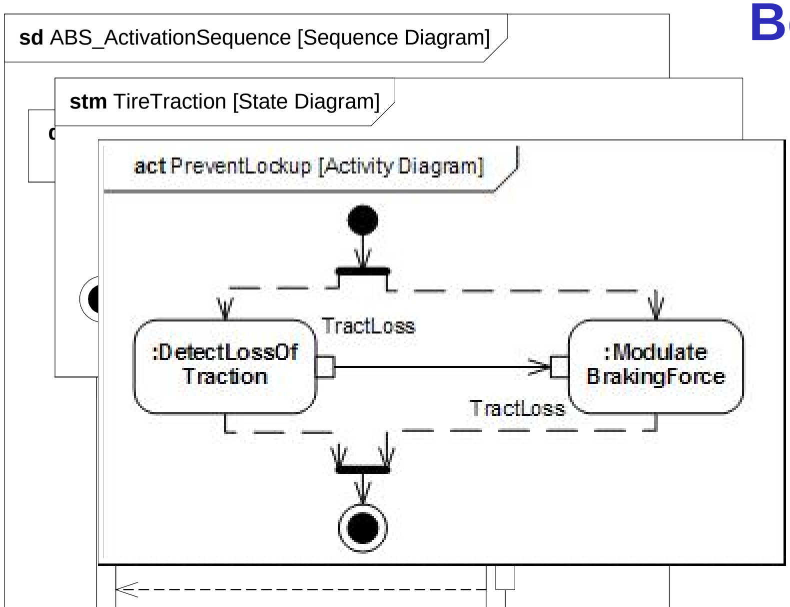


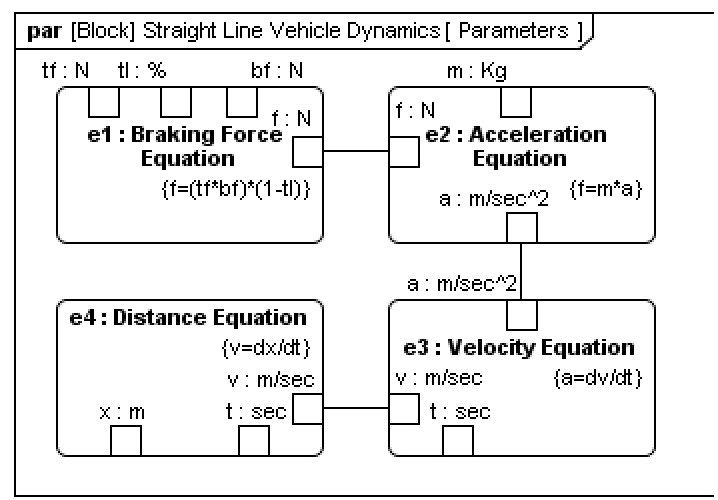
Behavior

interaction

state machine

activity



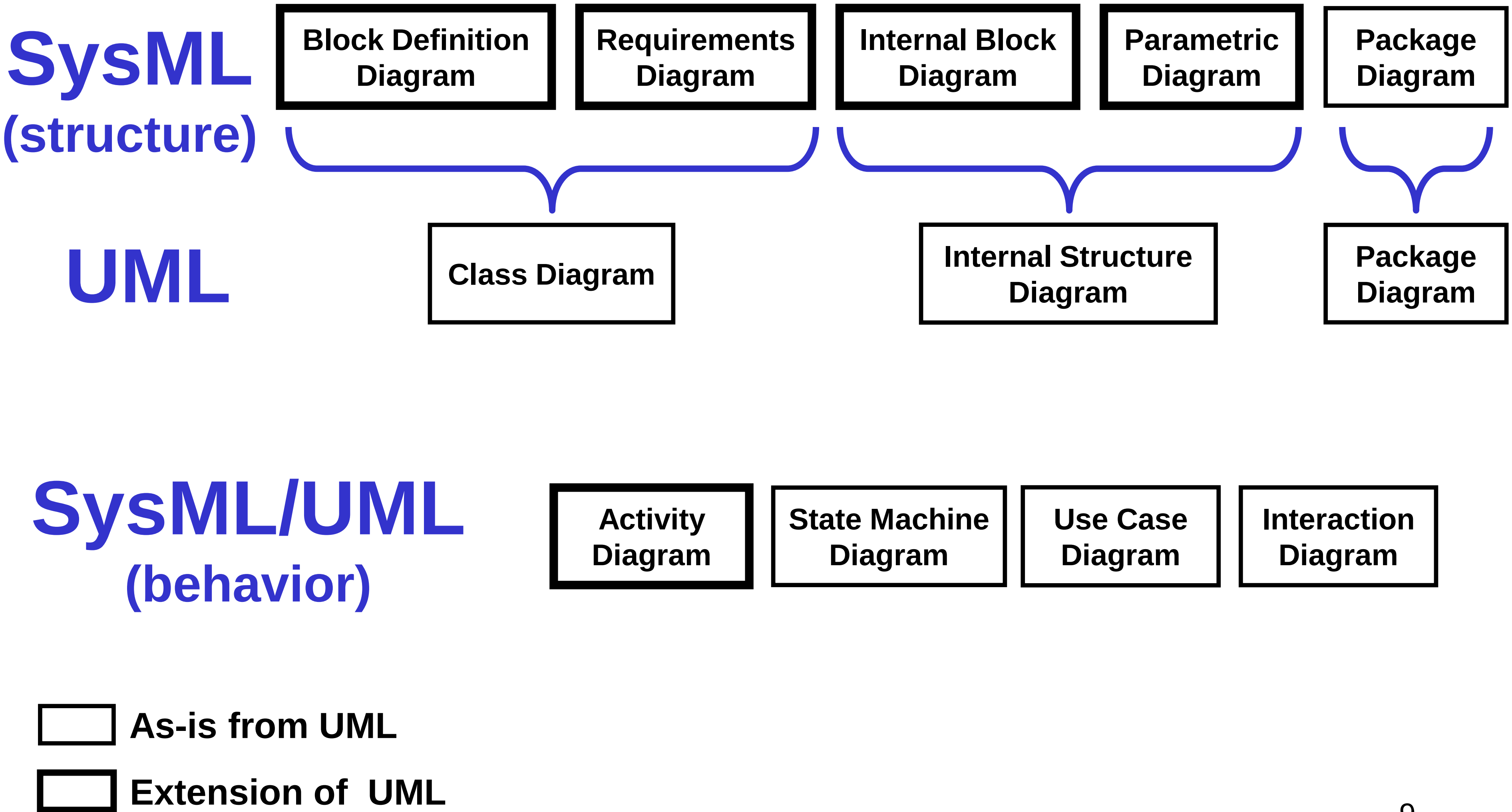


Parametrics

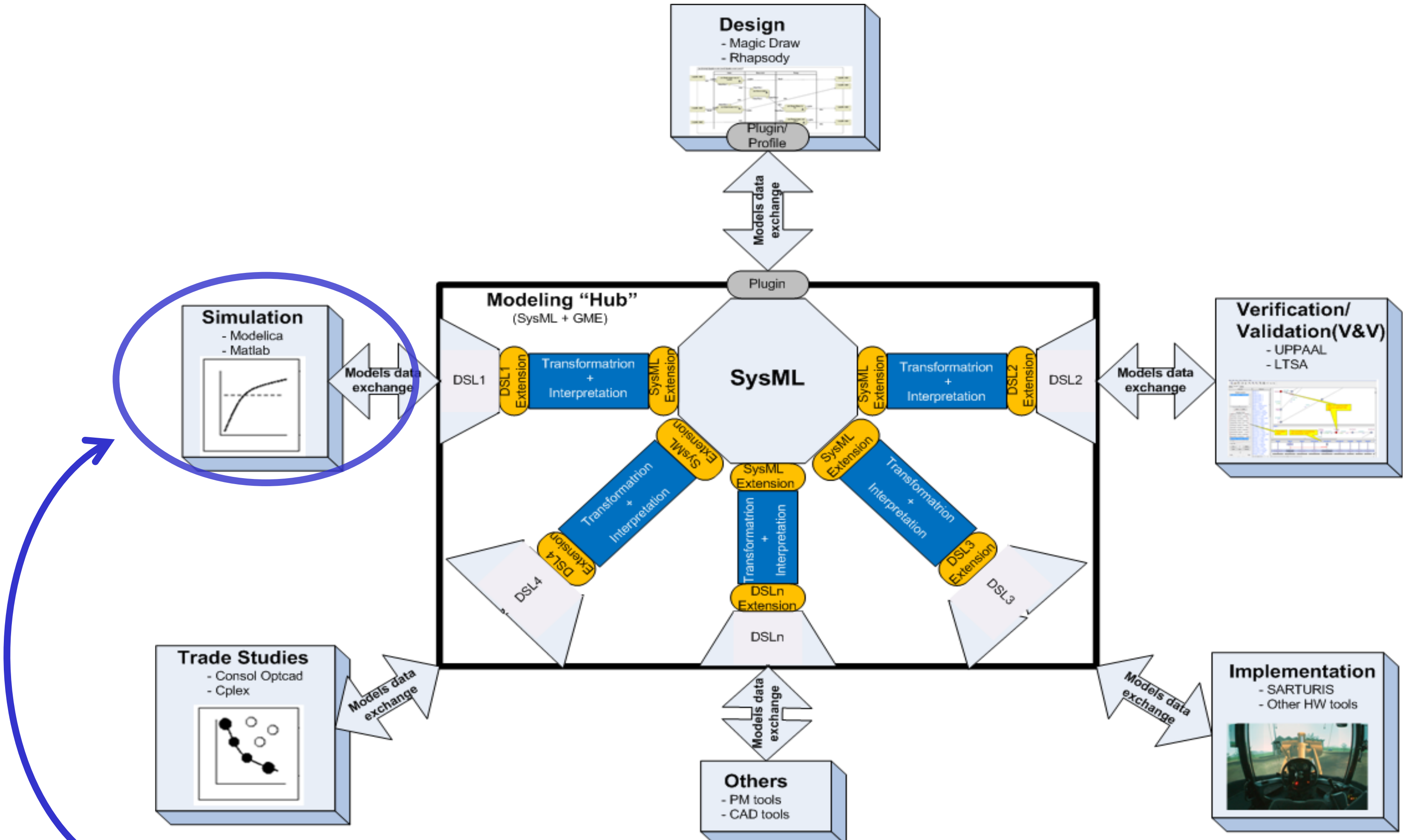
SysML extends the Unified Modeling Language (UML)

- UML is the most widely used graphical modeling language for software (also published by OMG).
- INCOSE chose to extend UML (and approach OMG) because
  - Modern systems/products usually have significant amounts of software in them.
  - Extending UML is a path to integrating engineering and software development.
  - Software modeling in UML has many commonalities with systems engineering modeling.

SysML/UML Diagrams



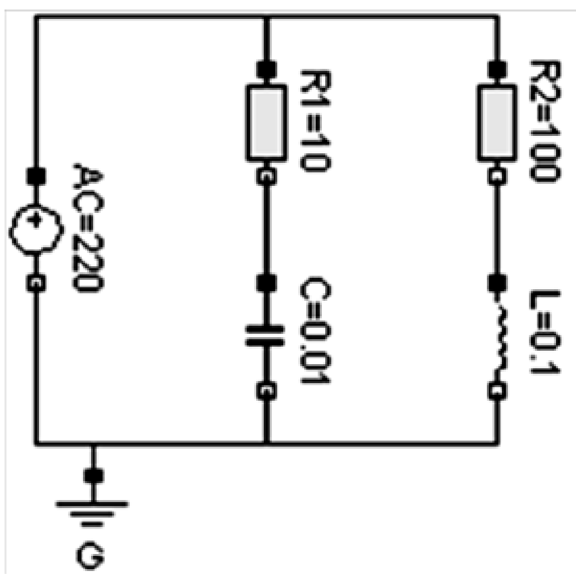
SysML as Hub for Engineering

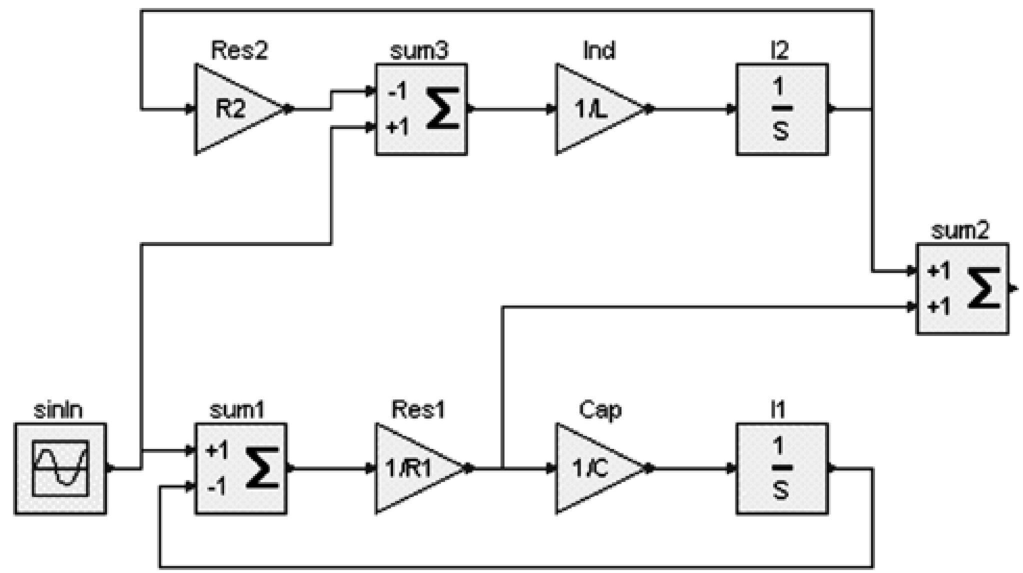


- **Focus of this talk is integration with solver-based simulation.**

Solver-based Simulators

- Solver-based simulators have user interfaces similar to modeling tools.

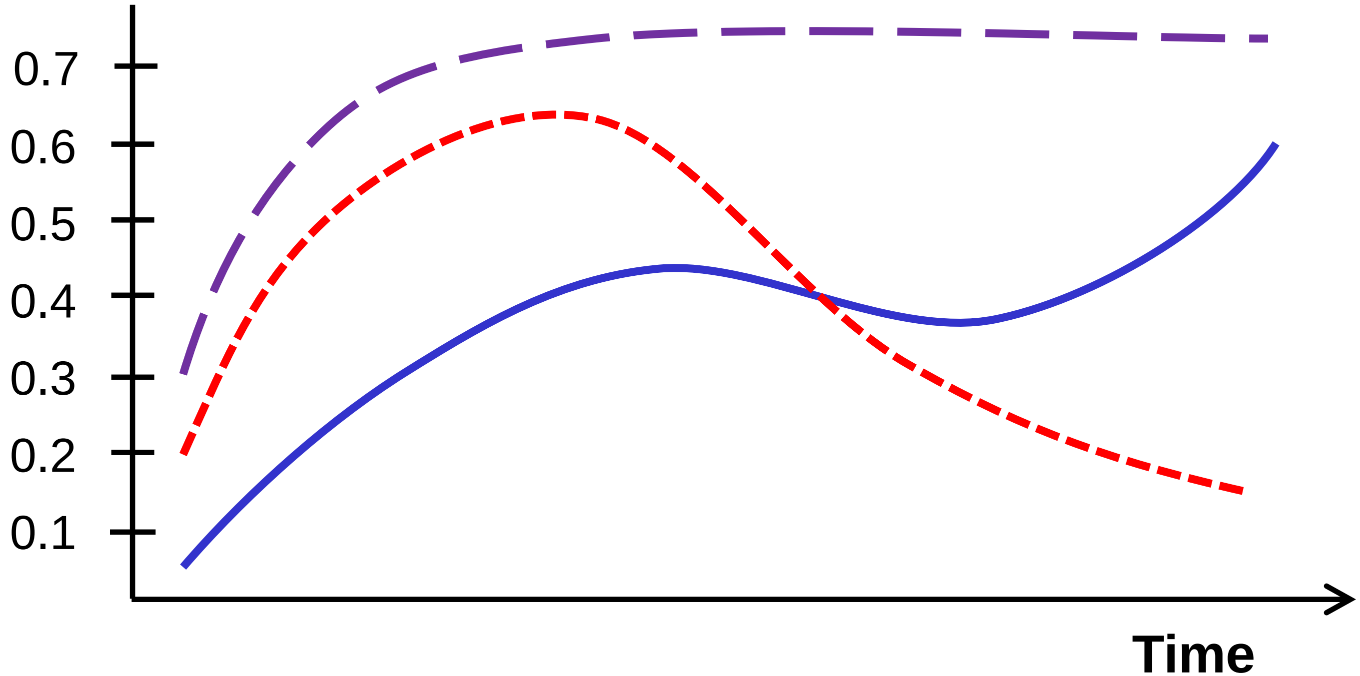




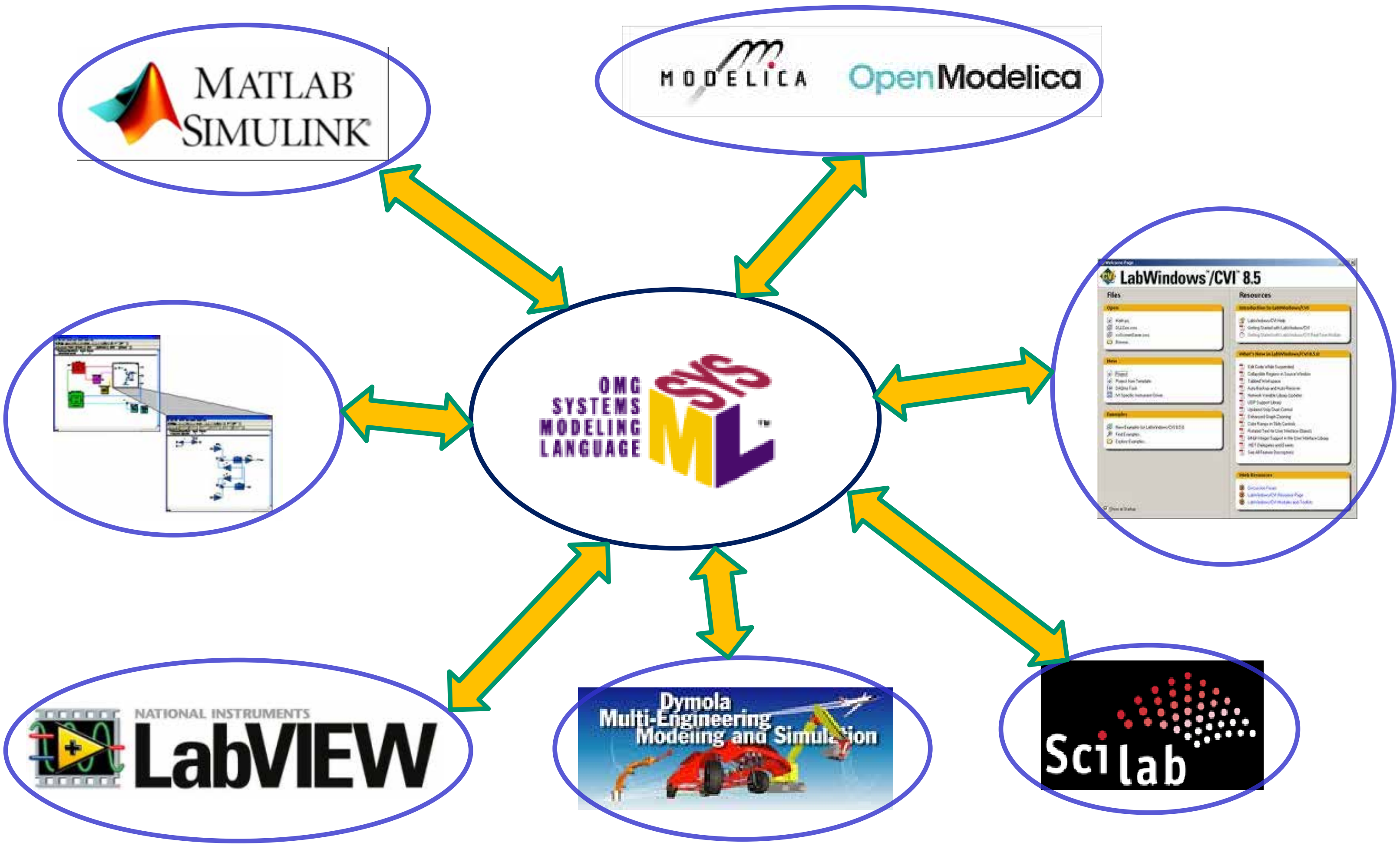
- But the tools treat these as equations rather than physical things.

Solver-based Simulators

- **Generate differential equations from diagrams and incrementally solve them to give values of variables over time.**

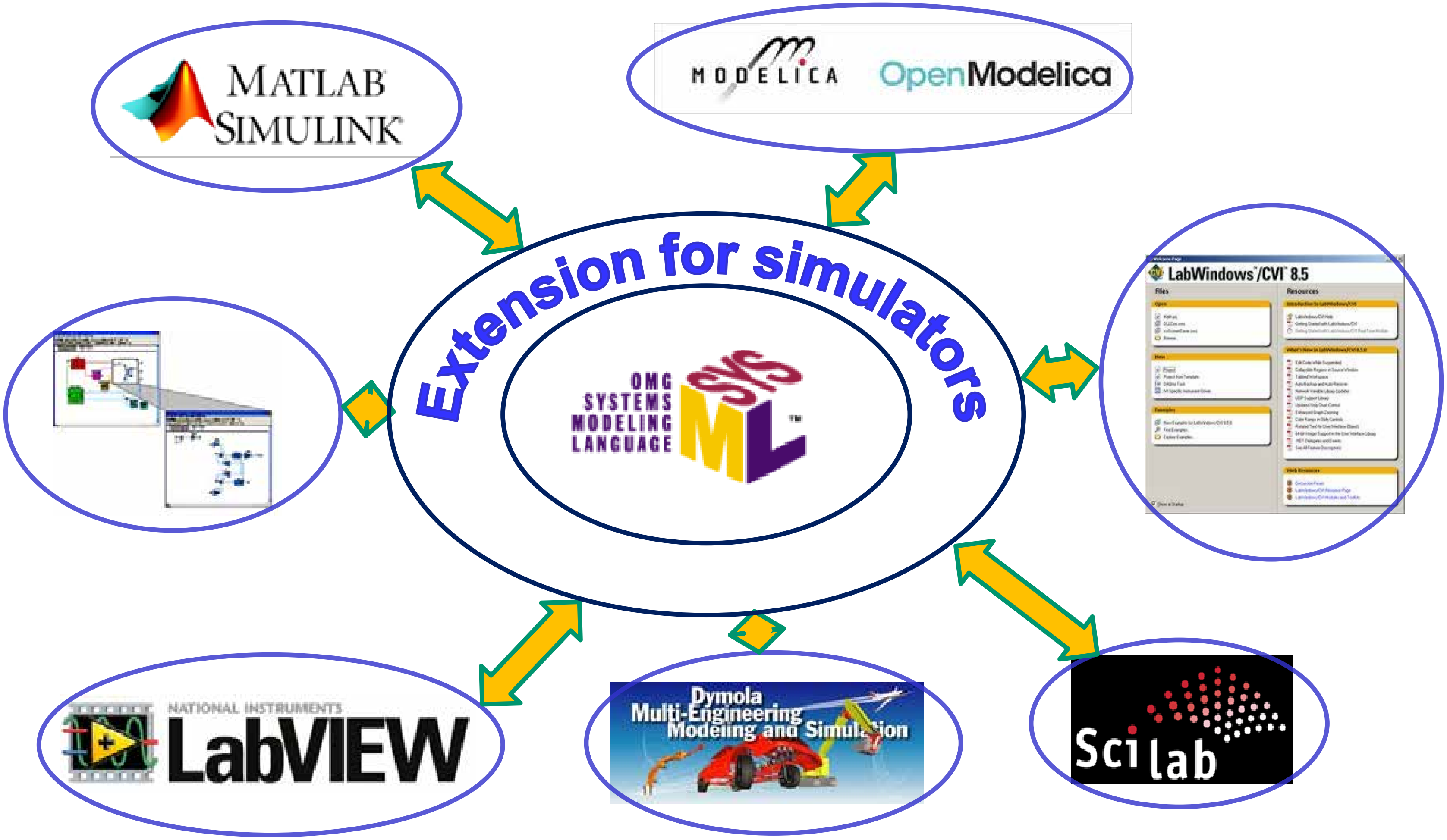


SysML Hub for Simulators



- **Integration supported by different profiles for each simulator.**

Reduce Specialized Profiles



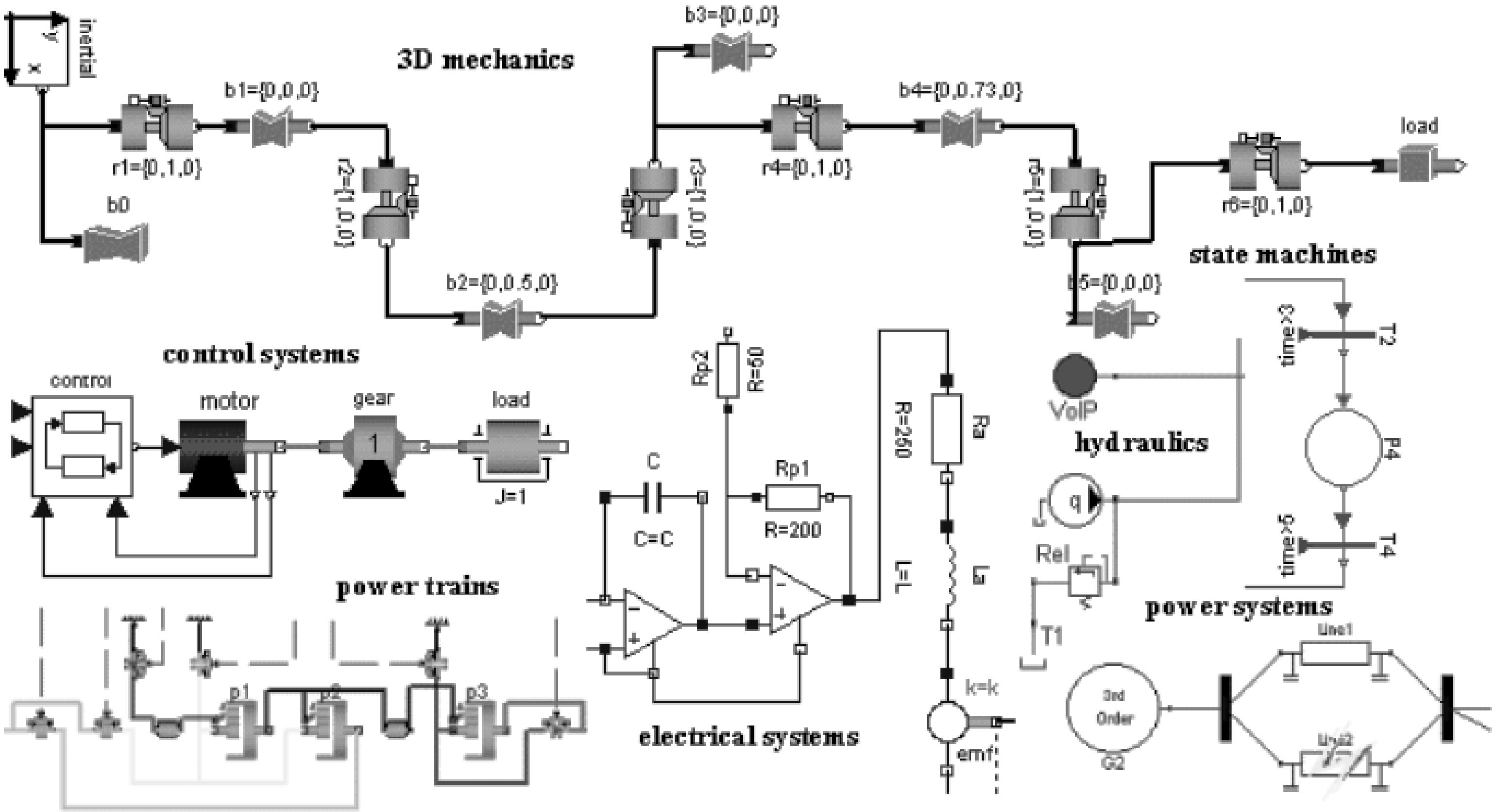
- Extend SysML with a general simulation profile.

Overview

§ Motivation and approach
§ **Dynamic simulation overview**
§ SysML extension
§ Detailed example
§ Transforming to simulation formats
§ Summary

Multiple Engineering Disciplines

- Generally, solvers use the same numerical algorithms for all the engineering disciplines.



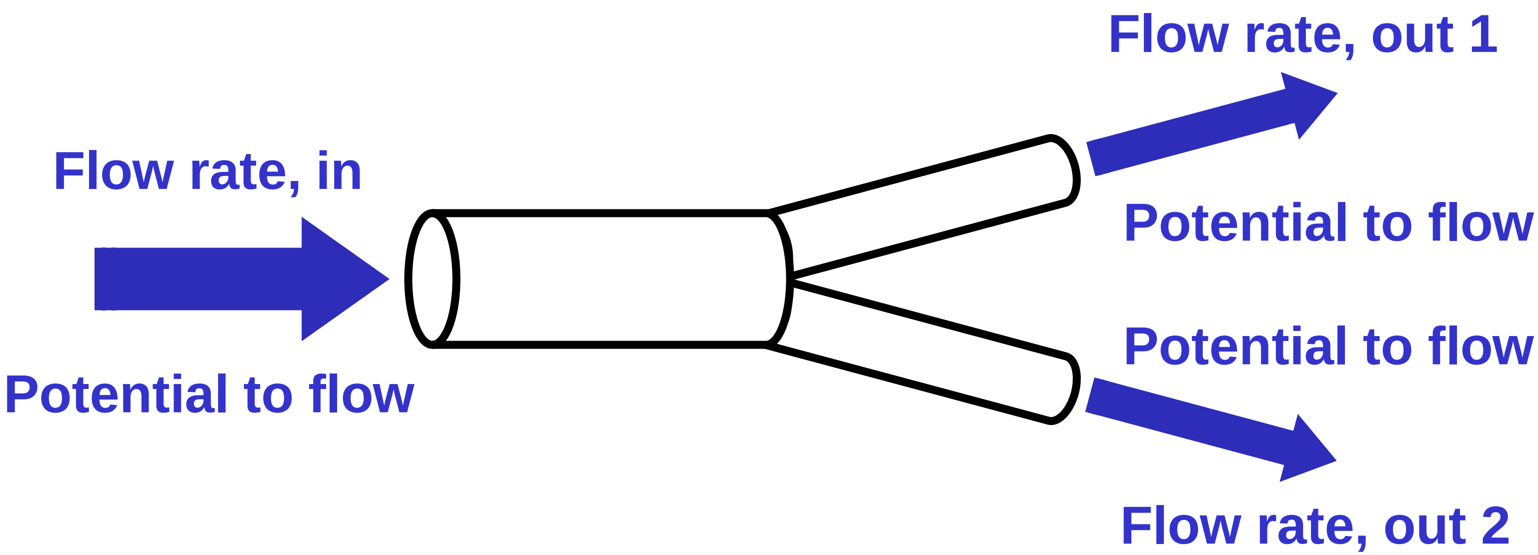
Multiple Engineering Disciplines

- **Possible because of commonality of underlying physics.**

Domain	What is flowing	Flow rate	Potential to flow
Electrical	Charge	Current	Voltage
Mechanics, translational	Momentum	Force	Velocity
Mechanics, angular	Angular momentum	Torque	Angular velocity
Hydraulics	Volume (uncompressable fluid)	Volumetric rate	Pressure
Thermal	Heat energy	Heat flow rate	Temperature

Conservation Laws

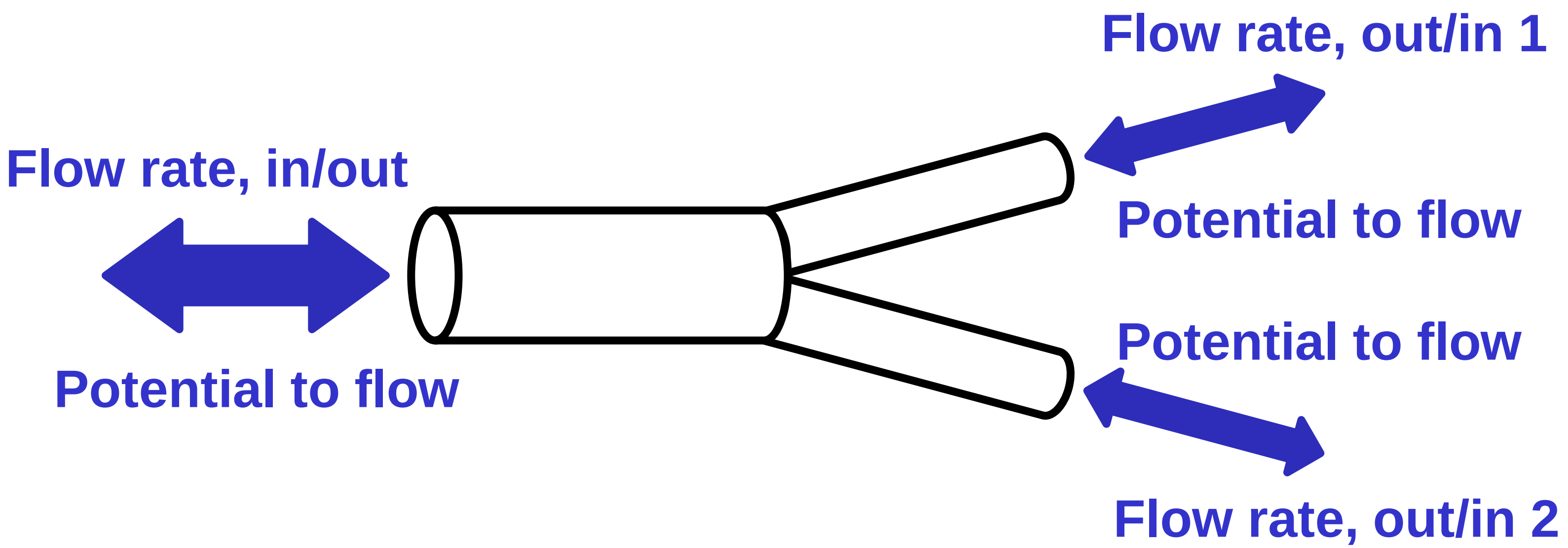
- Rates of flow follow conservation laws, potentials to flow do not.



- FR out 1 + FR out 2 = FR in
- Potential to flow is the same on all ends.

Conservation Laws

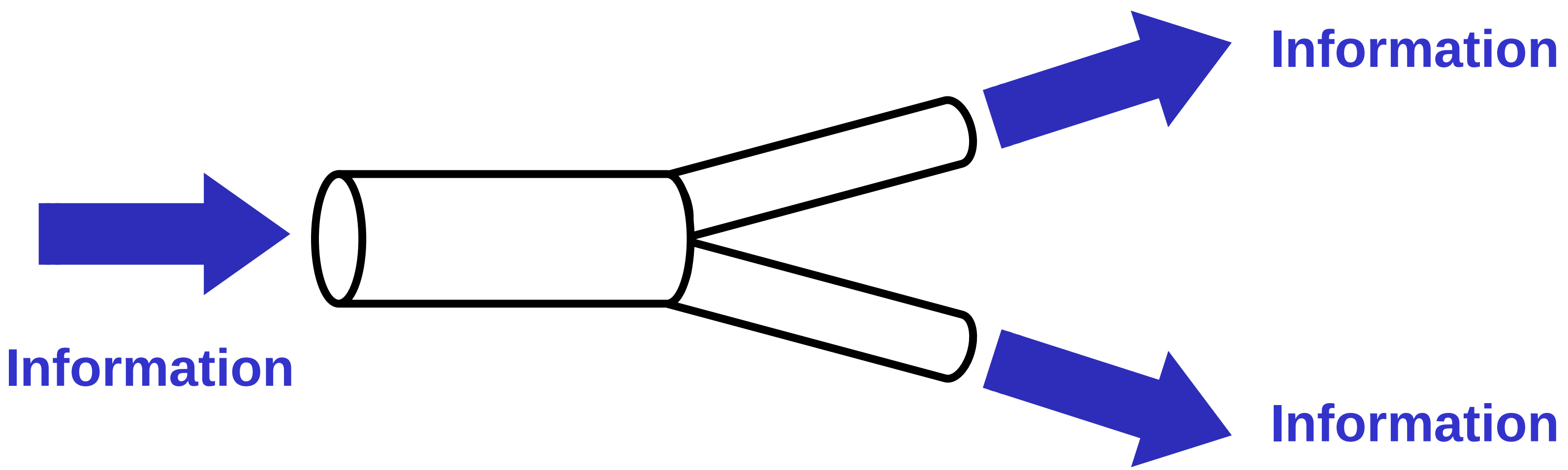
- Flow rates can be in either direction (postive or negative).



- FR out/in 1 + FR out/in 2 = FR in/out
- Potential to flow is the same on all ends.

Simulating Information Flow

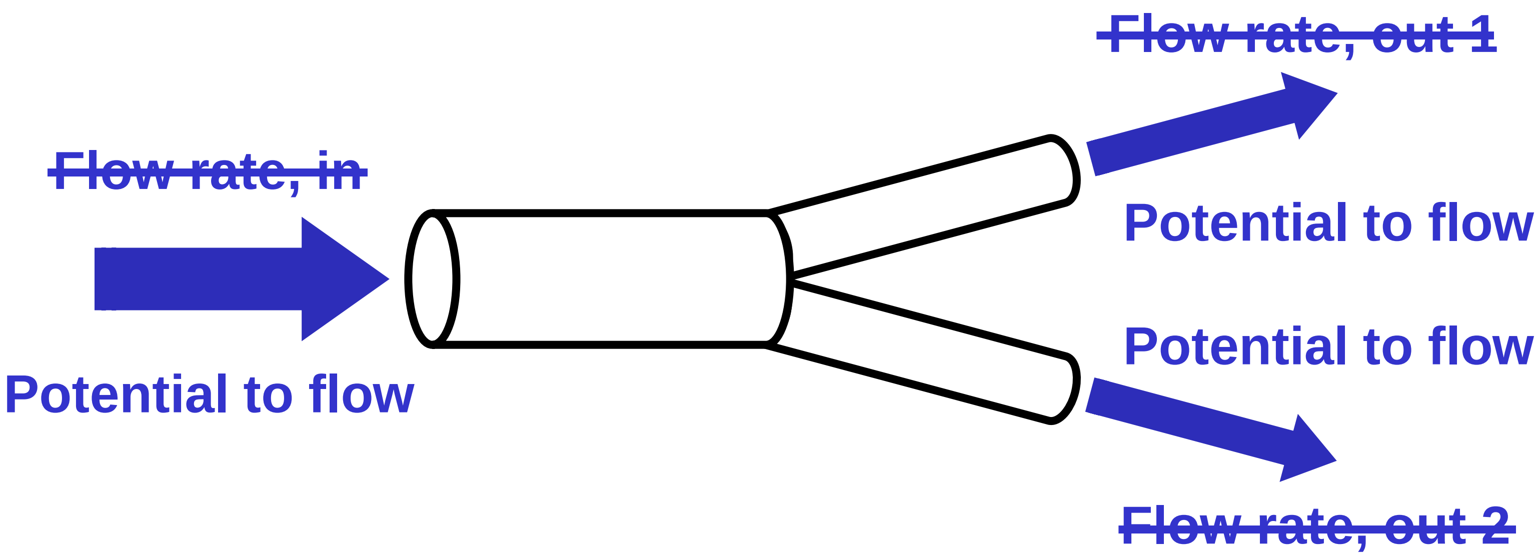
- **Information flow does not follow conservation laws**
  - **Information can be copied.**
  - **Simulated as potential to flow (signals).**



- **Information is the same on all ends.**

Simulator Constraints

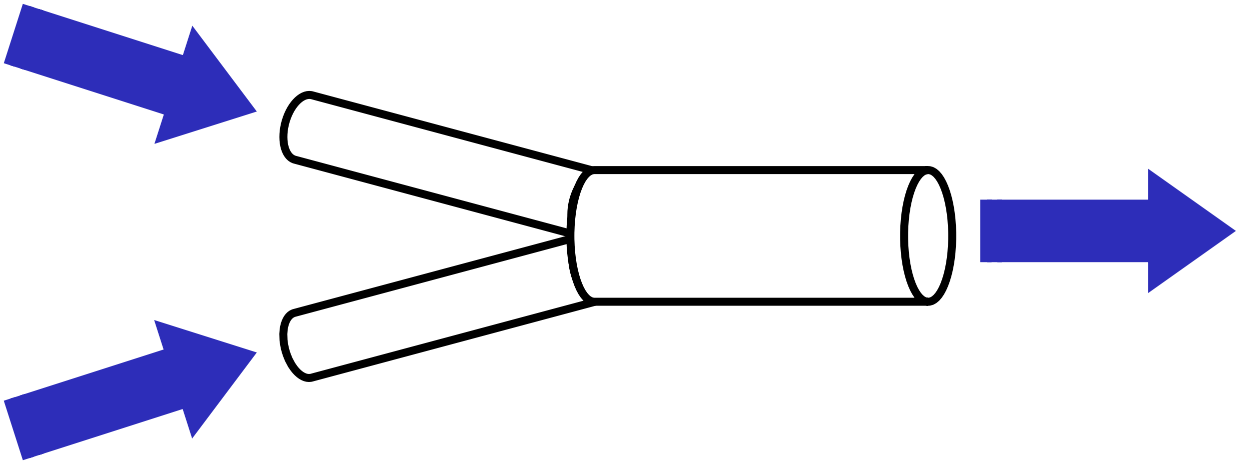
- Rates of flow cannot be simulated on unidirectional flows.



- ~~FR out 1 + FR out 2 = FR in~~
- Potential to flow is the same on all ends.

Simulator Constraints

- Unidirectional flows cannot be merged.



- They can be split (reverse of above).
- Bidirectional flows can be merged and split.

Overview

- Motivation and approach
- Dynamic simulation overview
- **SysML extension**
- Detailed example
- Transforming to simulation formats
- Summary

Integration with SE Modeling

- **Systems engineering models and simulators are concened with overlapping aspects of flow.**

Domain	Kind of item flowing
Electrical	Charge
Mechanics, translational	Momentum
Mechanics, angular	Angular momentum
Fluid	Volume (uncompressable fluid)
Thermal	Heat energy

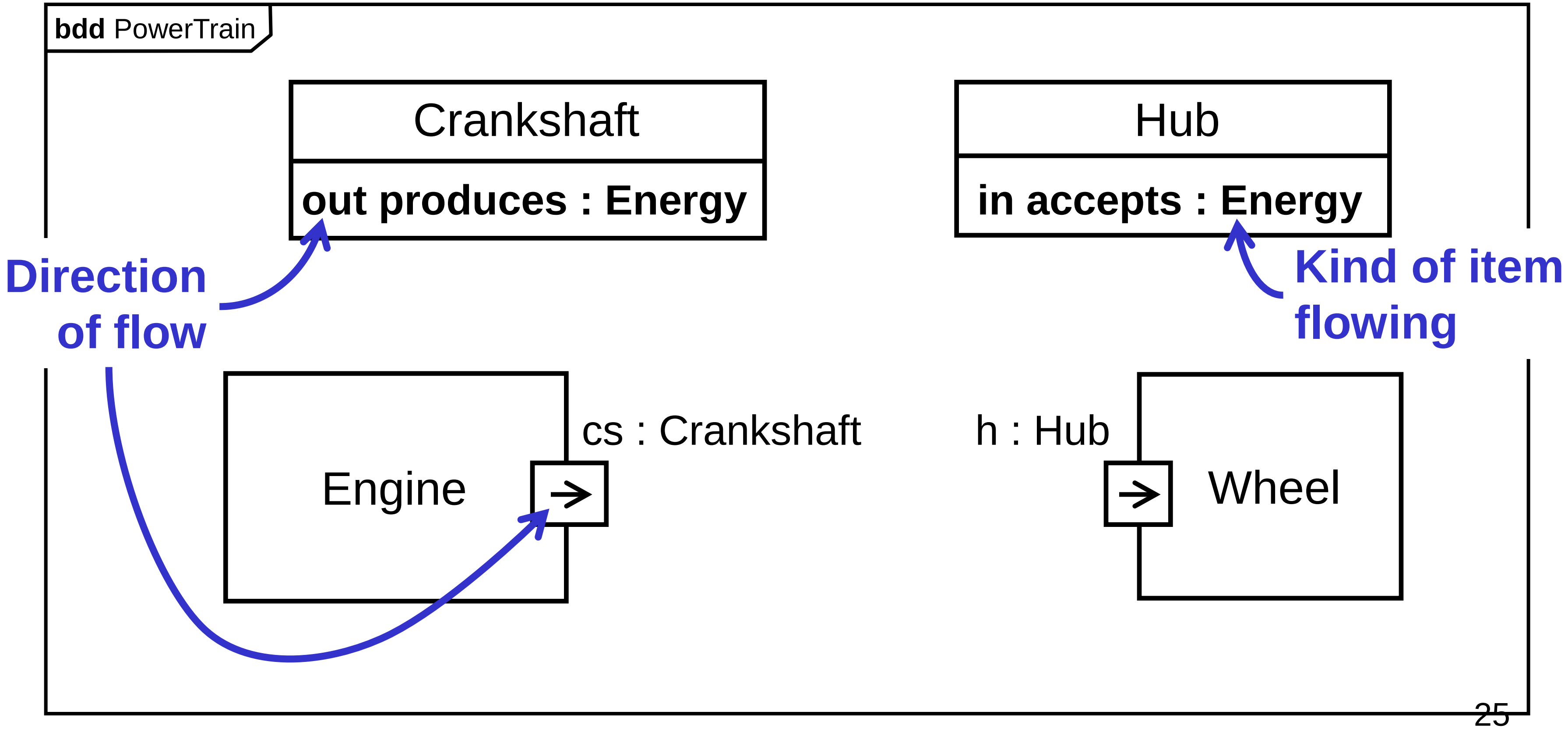
Direction of flow

Flow Rate	Potential to flow
Current	Voltage
Force	Velocity
Torque	Angular velocity
Volumetric rate	Pressure
Heat flow rate	Temperature

Systems Engineering **Dynamic Simulators**

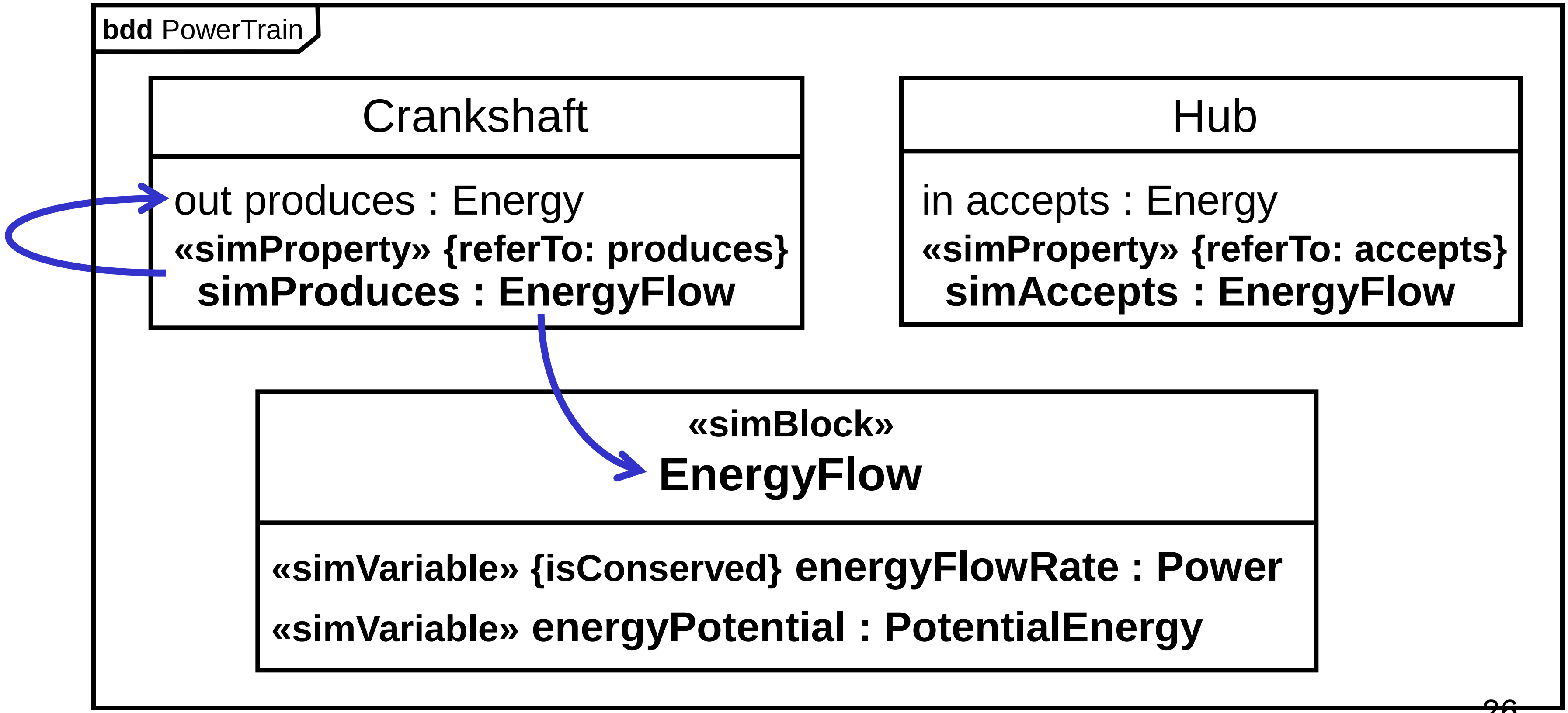
Flow Properties in SysML

- **Specify what is flowing and in which direction.**

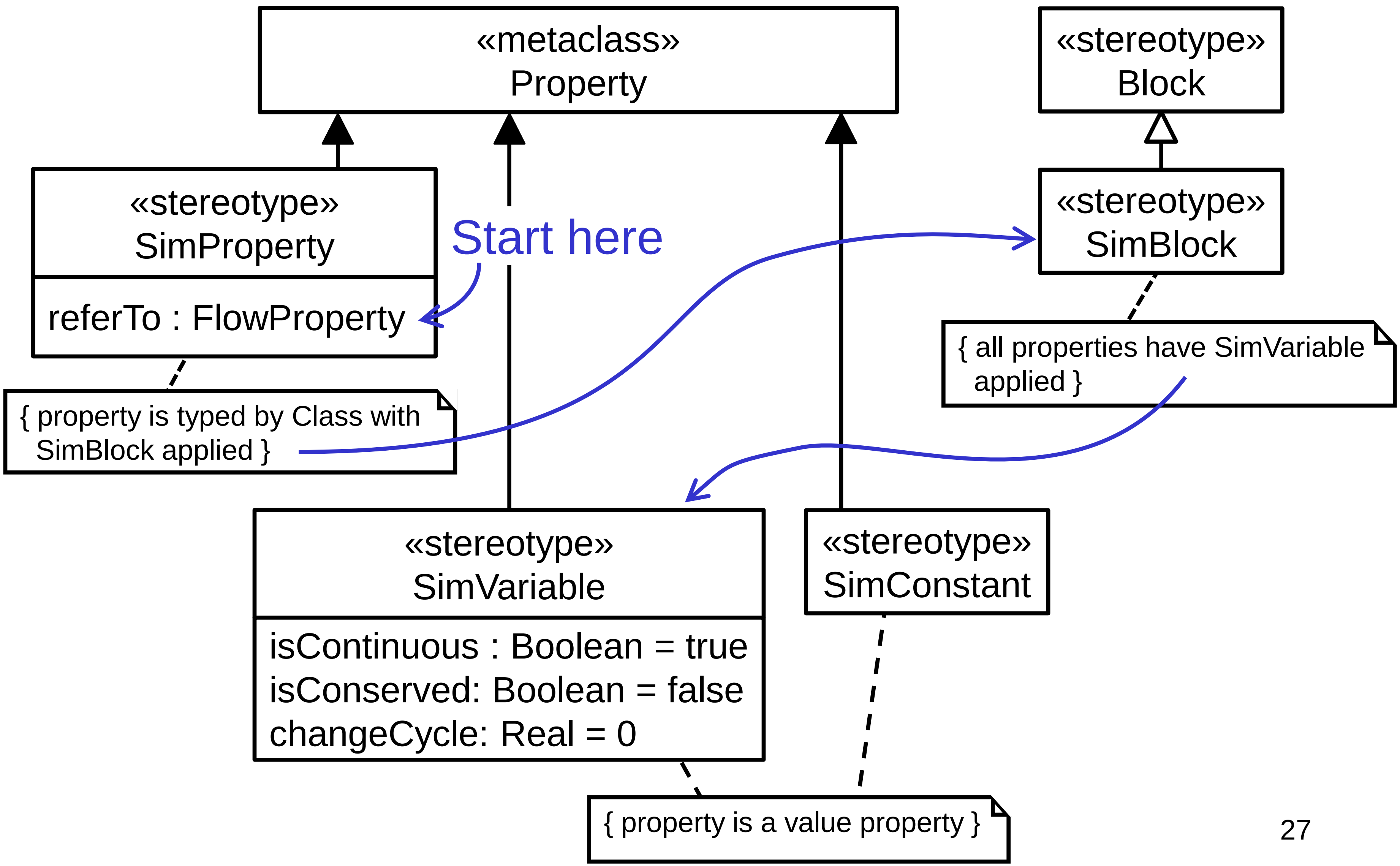


Extending SysML

- Bring flows and potentials into SysML for generating simulator input.



Stereotypes



Conservation and Directionality

- **SimBlocks for unidirectional flow properties (*in* or *out*) can only have non-conserved variables (isConserved = false).**
- **Simblocks for bidirectional flow properties can have both conserved and non-conserved variables.**

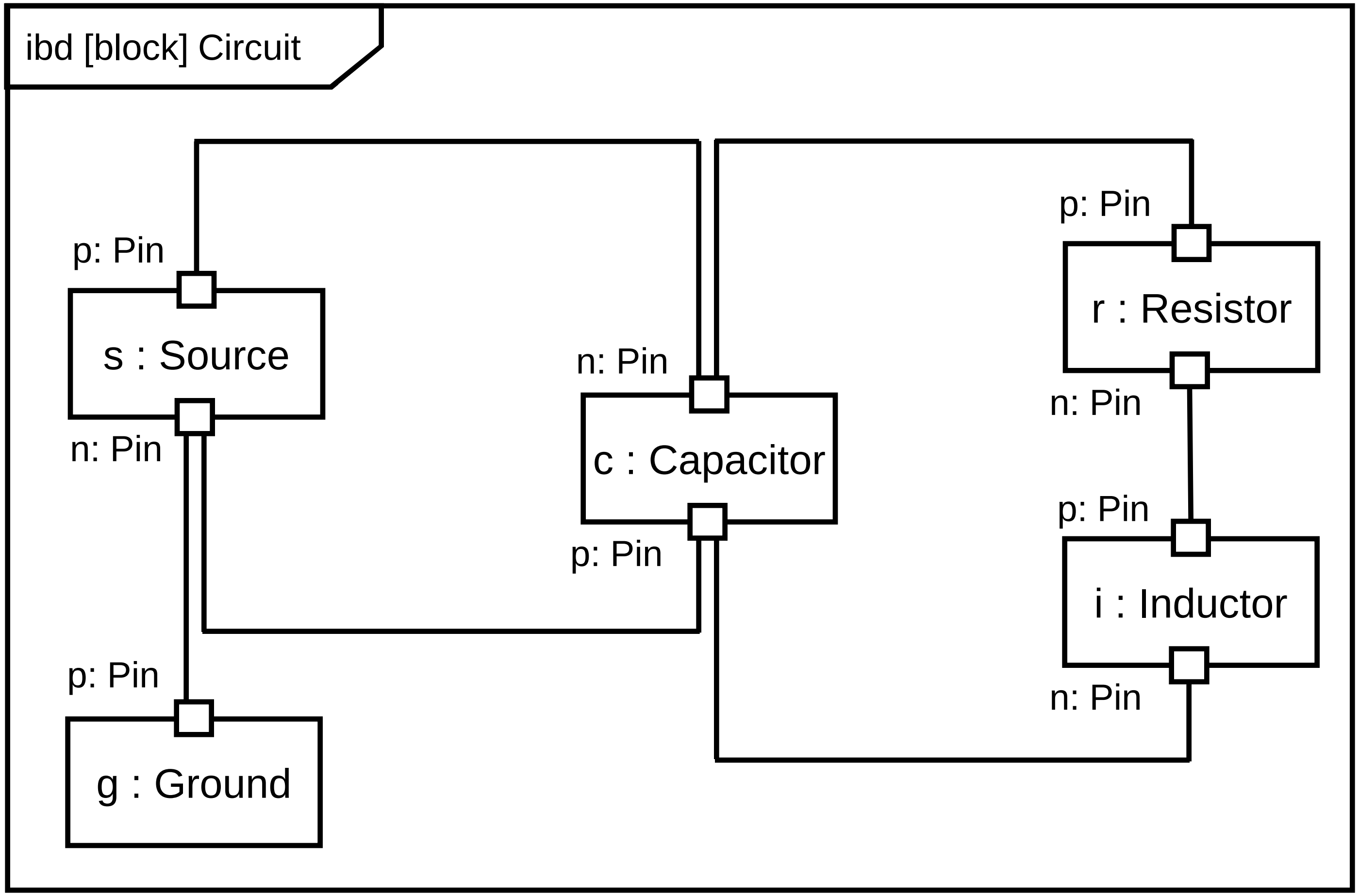
Connection Constraints

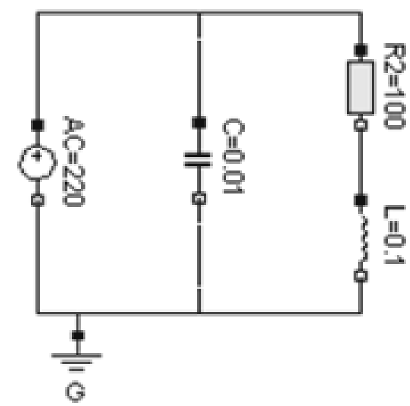
- SimBlocks of matching flow properties must either be the same or match exactly.
- *In* flow properties can be connected to no more than *out* flow property.
- *Out* flow properties can be connected to any number of *in* flow properties.
- *Inout* flow properties aren’t constrained in linkage number.

Overview

- Motivation and approach
- Dynamic simulation overview
- SysML extension
- **Detailed example**
- Transforming to simulation formats
- Summary

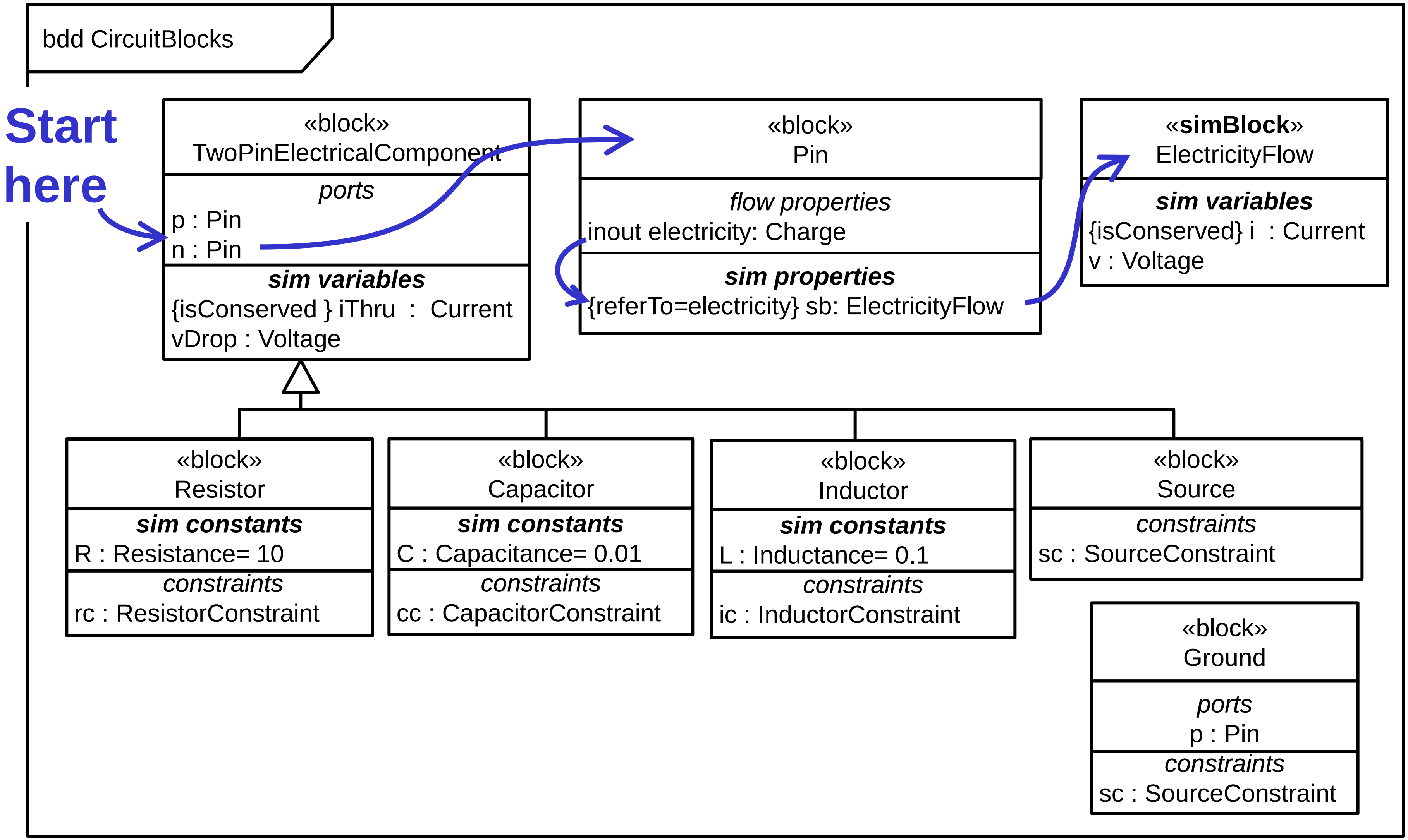
Example (Graphics)



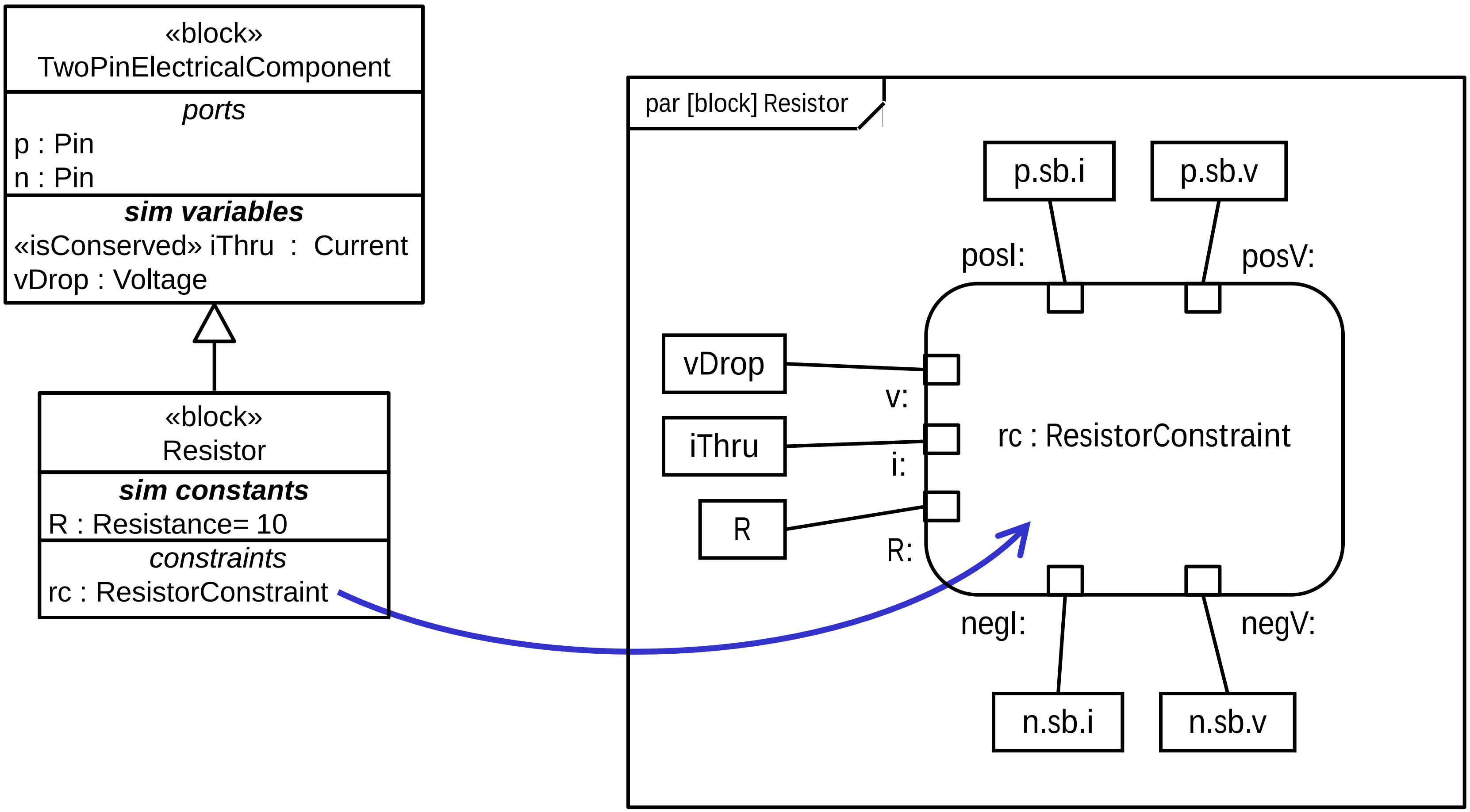


- N-ary electrical connections broken into binary SysML connectors.

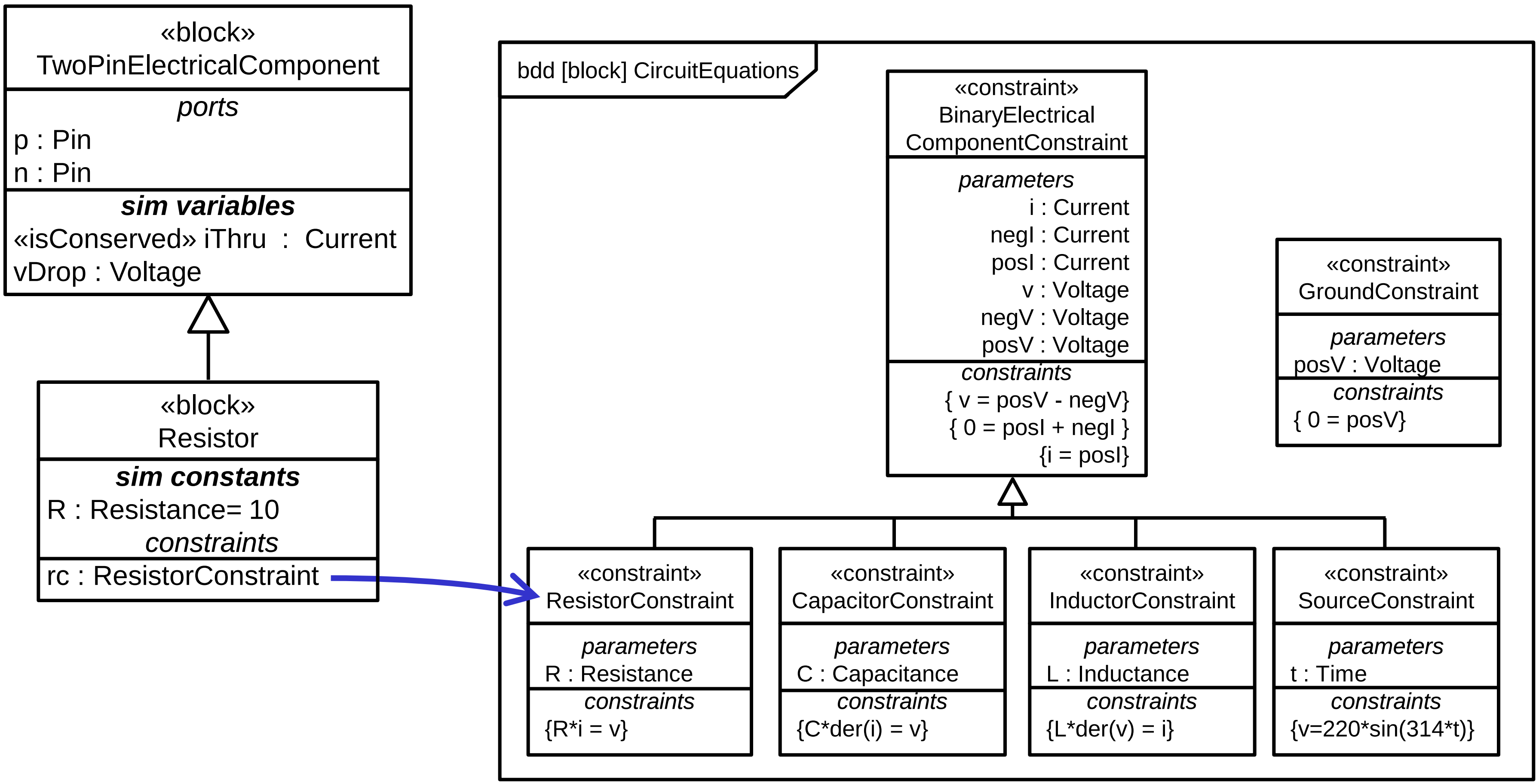
Example (Extensions)



Example (Constraint Blocks)



Example (Constraints)



- **Specifying mathematical equations.**

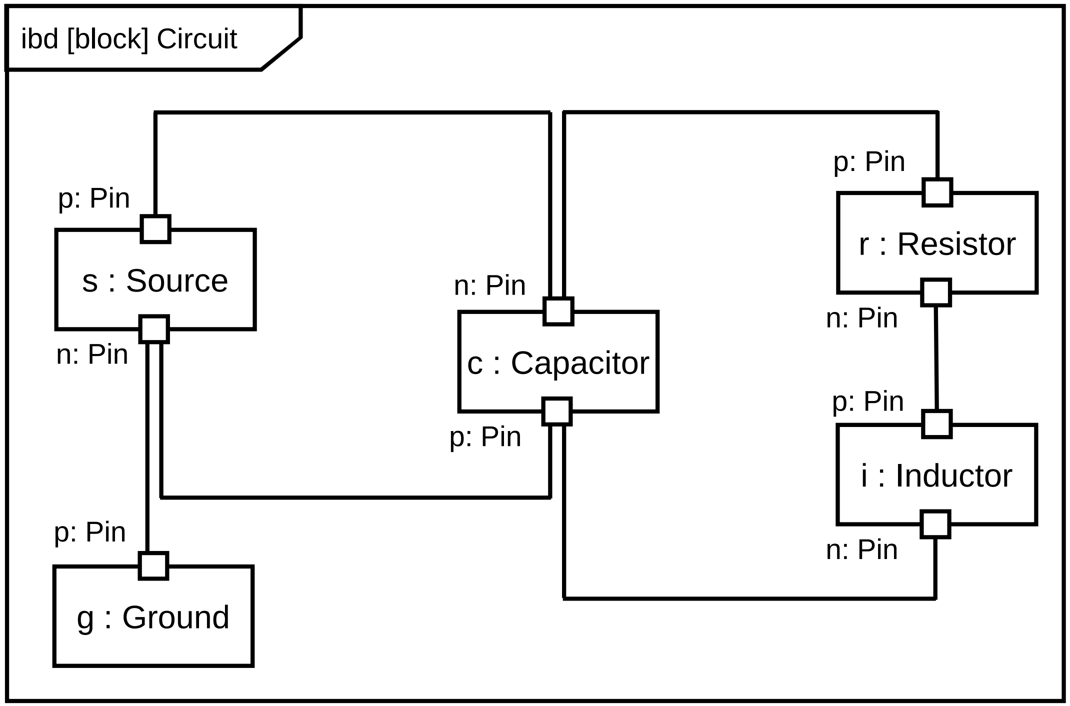
Overview

- Motivation and approach
- Dynamic simulation overview
- SysML extension
- Detailed example
- **Transforming to simulation formats**
- Summary

Terminology Mapping

SysML+	**Modelica**	**Simulink / Simscape**
Block without internal structure	Model without connections	BlockType / Component
Block with internal structure	Model with connections	System / Component
SimBlock (referring to a flow prop)	**Connector**	**Library elements**
Variables On SimBlocks	**Variables**	**Ports / Variables**
Connector	Connection/Equation	Line/Connection
Constraint block	Equation	S-Function / Equation

Mapping Internal Structure to Modelica

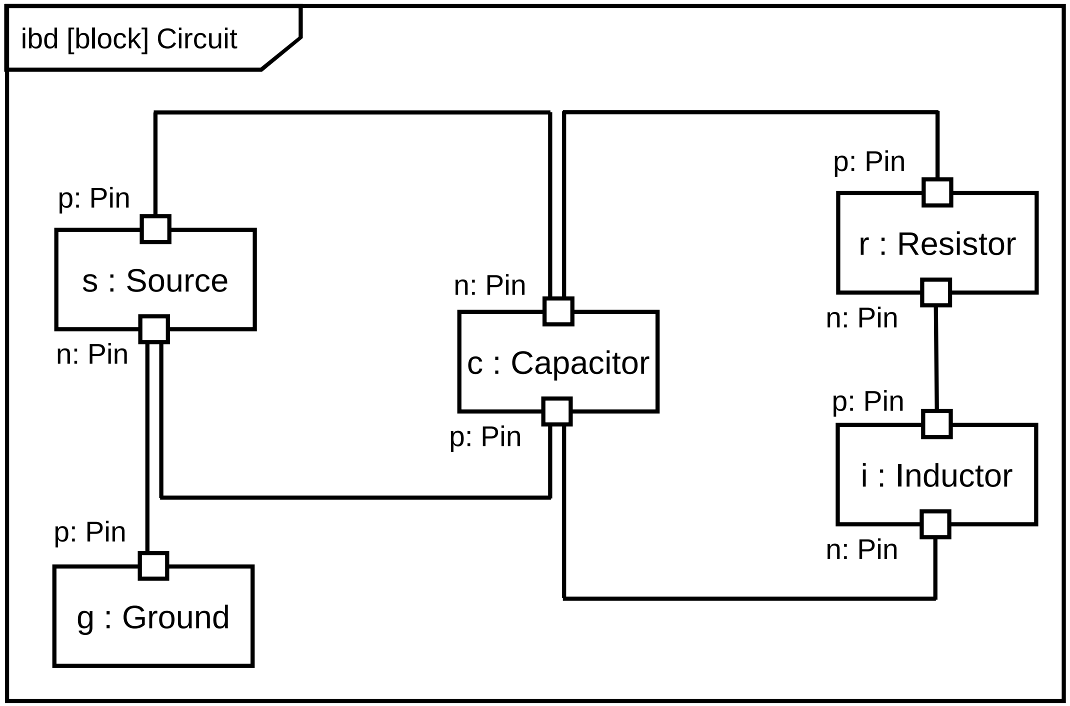



```
model circuit
    Resistor  r(R=10);
    Capacitor c(C=0.01);
    Inductor  i(L=0.1);
    Source s;
    Ground g;
equation
    connect (s.p, c.n);
    connect (c.n, r.p);
    connect (r.n, i.p);
    connect (i.n, c.p);
    connect (c.p, s.n);
    connect (s.n, g.p);
end circuit;
```

SysML

Modelica

Mapping Internal Structure to Simscape



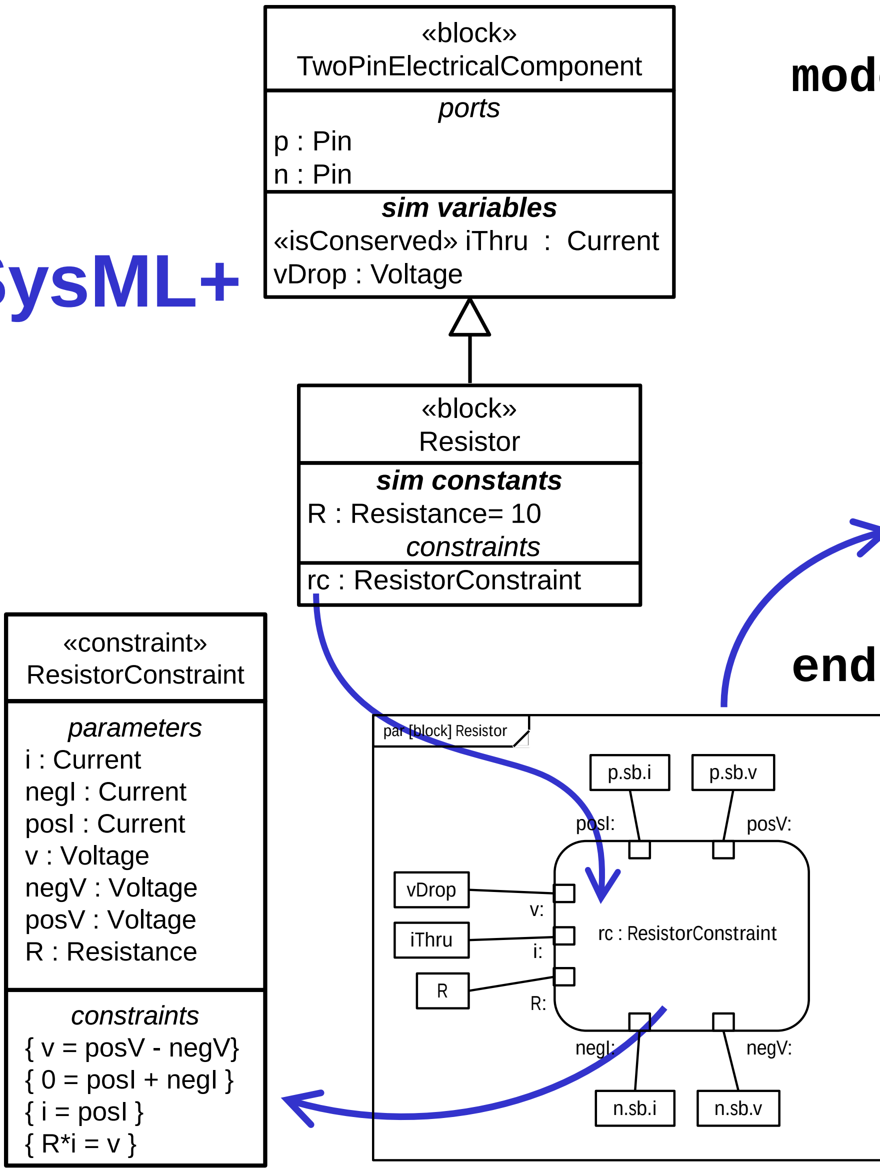
SysML

```
component circuit
 components
   r = Resistor(R=10);
   c = Capacitor(C=0.01);
   i = Inductor(L=0.1);
   S = Source;
   G = Ground;
 end
 connections
   connect (s.p, c.n);
   connect (c.n, r.p);
   connect (r.n, i.p);
   connect (i.n, c.p);
   connect (c.p, s.n);
   connect (s.n, g.p);
 end
end
```

Simscape

Mapping Constraints to Modelica

SysML+



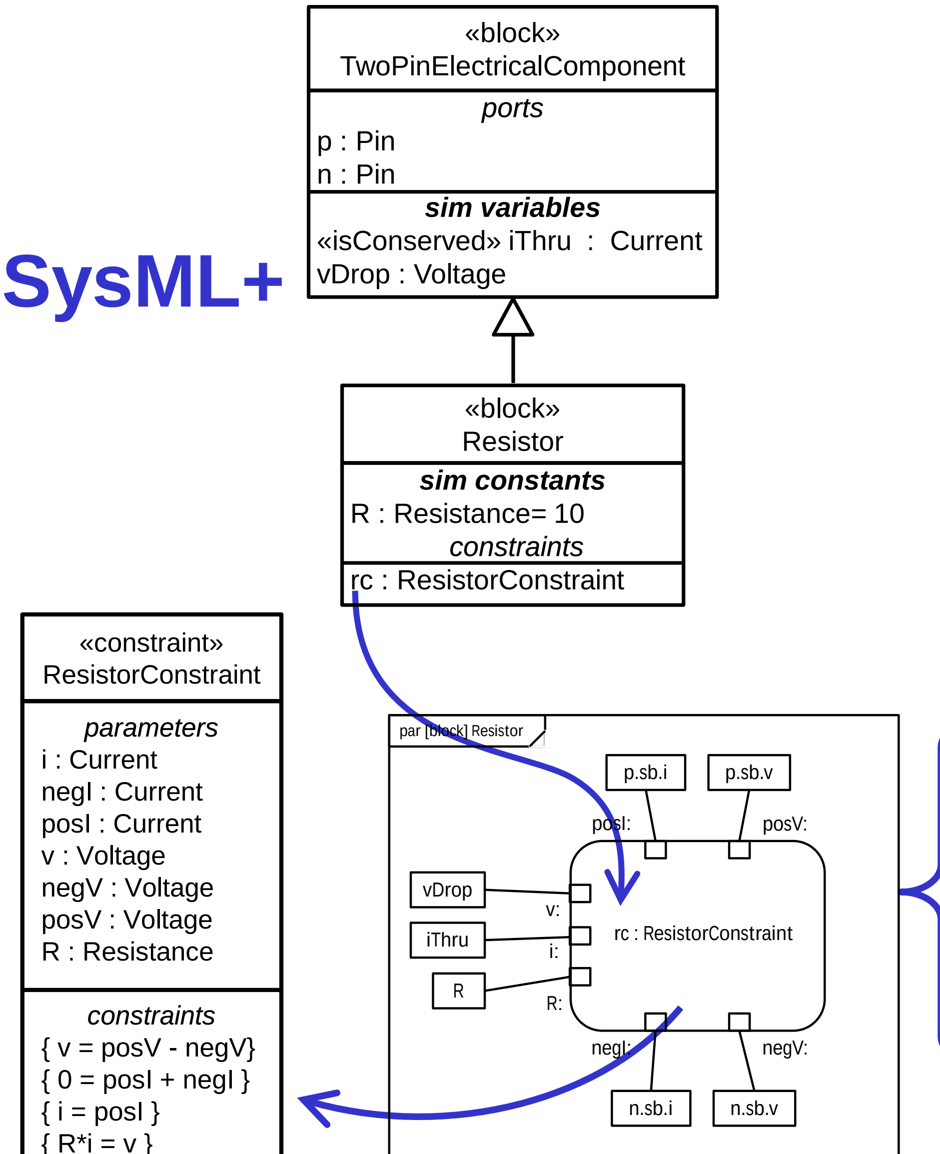

```
model Resistor “Electrical resistor"
    Pin p,n;
    flow Current iThru;
    Voltage vDrop;
    parameter Real R(unit="Ohm")
        "Resistance";
    equation
        vDrop = p.v - n.v;
        0 = p.i + n.i;
        iThru = p.i;
        R*iThru = vDrop;
end Resistor
```

Modelica

- **Equations in simulator use variable names from model *after* binding.**

Mapping Constraints to Simscape

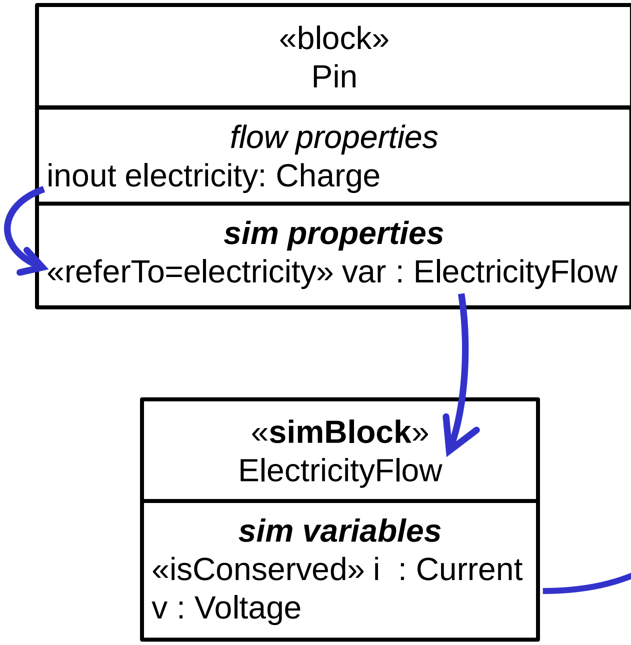
SysML+




```
component “Electrical Resistor“
 nodes
    p = foundation.electrical.
                         electrical;
    n = foundation.electrical.
                         electrical;
  end
 variables
    iThru = { 0, 'A' };
    vDrop = { 0, 'V' };
 end
 parameters
    R = { 1, 'Ohm' };
 end
 function setup
    across( vDrop, p.v, n.v );
    through( iThru, p.i, n.i );
 end
 equations
    R*iThru == vDrop;
 end
end
```

Simscape

Mapping SimBlocks to Modelica

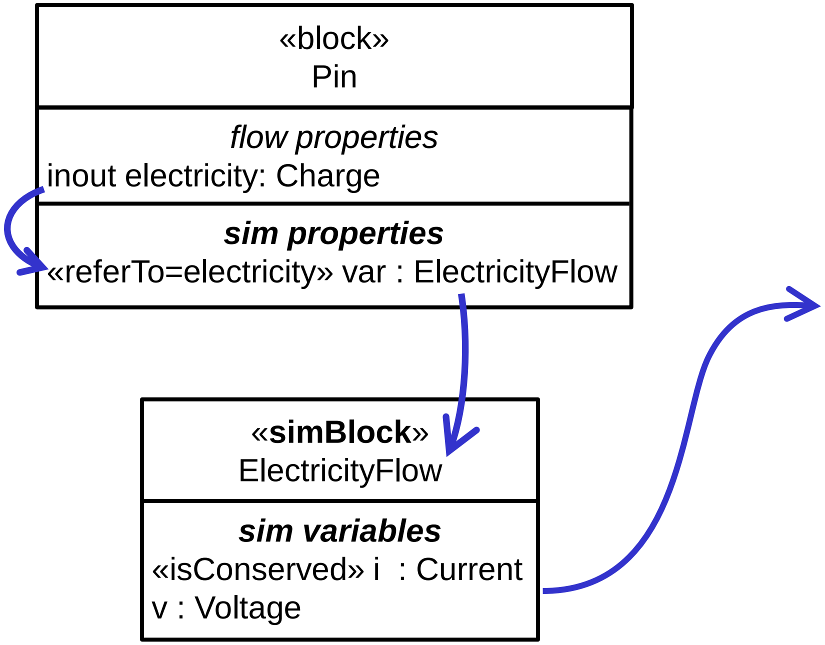



```
connector Pin
  flow Current i;
  Voltage      v;
end EF;
```

SysML+ **Modelica**

- **Pin in simulator has properties of SimBlocks**
  - **Flow properties used only to determine direction (“causality”) in usages of SimBlocks.**

Mapping SimBlocks to Simscape



We can't find how these elements are specified, but they are referred to in user models, see later slides.

SysML+ **Simscape**

- **Pin in simulator is only the SimBlock**
  - **Flow properties used only to determine direction ("causality") in usage of SimBlocks.**

Overview

- Motivation and approach
- Dynamic simulation overview
- SysML extension
- Detailed example
- Transforming to simulation formats
- **Summary**

Summary

- Goal is to reduce size and complexity of simulator-specific profiles.
  - by reusing and extending SysML.
- SysML concerned with flow direction (input/output) and kind of things flowing.
- Simulators are concerned with flow direction, potential, and rate.
- Extend SysML with rate, potential, and other aspects of simulated flow.
- Use extended SysML to generate simulator-specific files.

More Information

- **An Analysis of Solver-Based Simulation Tools**
  - **Survey of solver-based simulators**
  - **(http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909924)**
- **Modeling Methodologies and Simulation for Dynamical Systems**
  - **Describes two ways simulators are used.**
  - **(http://nvlpubs.nist.gov/nistpubs/ir/2012/NIST.IR.7875.pdf)**
- **SysML Extension for Dynamical System Simulation Tools**
  - **Covers a simulator-independent extension of SysML.**
  - **(http://nvlpubs.nist.gov/nistpubs/ir/2012/NIST.IR.7888.pdf)**