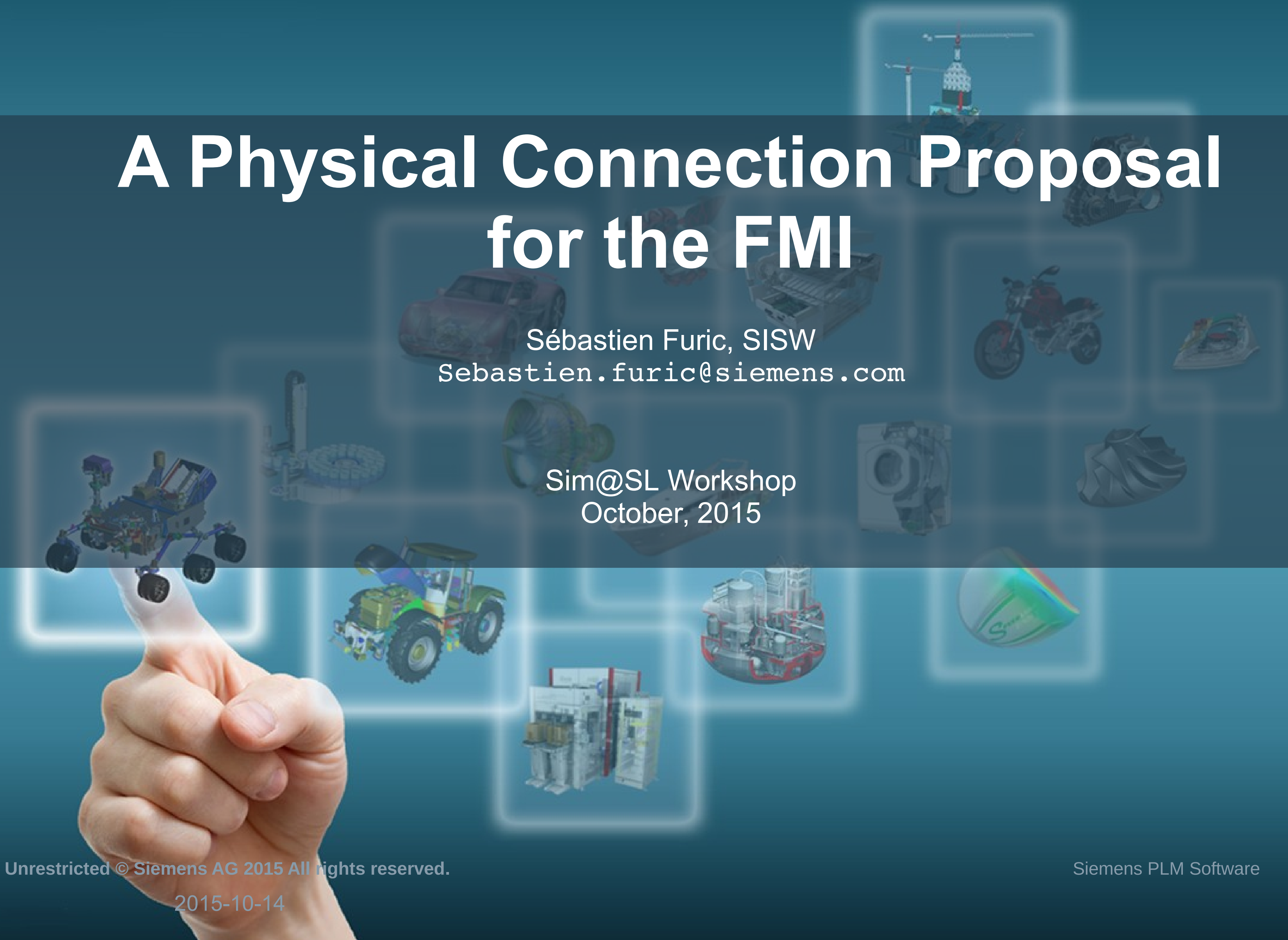
# A Physical Connection Proposal for the FMI

Sébastien Furic, SISW
Sebastien.furic@siemens.com

Sim@SL Workshop
October, 2015

Unrestricted © Siemens AG 2015 All rights reserved.

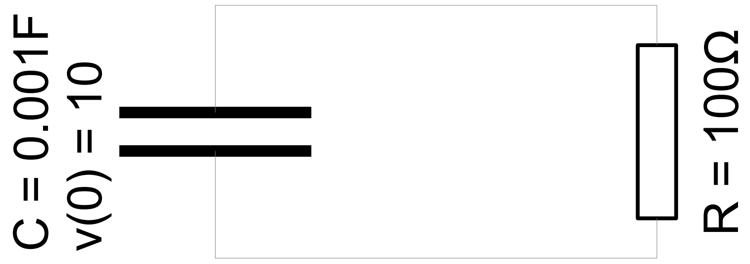


## Motivations

Unrestricted © Siemens AG 2015 All rights reserved.

## Are Block Diagrams Well Suited to Physical Modelling?

Consider this simple electrical model:



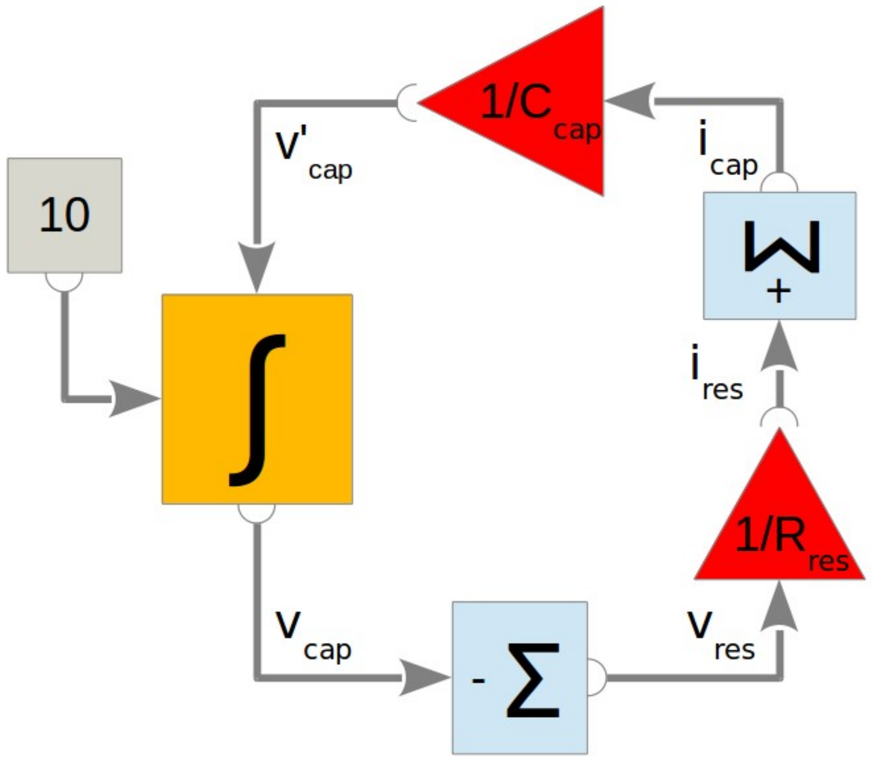
We will suppose in the following that its behaviour is governed by these equations:

$$\begin{cases} C_{cap} \cdot v'_{cap} = i_{cap} & \text{(capacitor's constitutive equation)} \\ v_{res} = R_{res} \cdot i_{res} & \text{(resistor's constitutive equation)} \\ i_{cap} = i_{res} & \text{(Kirchhoff's current law)} \\ v_{cap} + v_{res} = 0 & \text{(Kirchhoff's voltage law)} \end{cases}$$

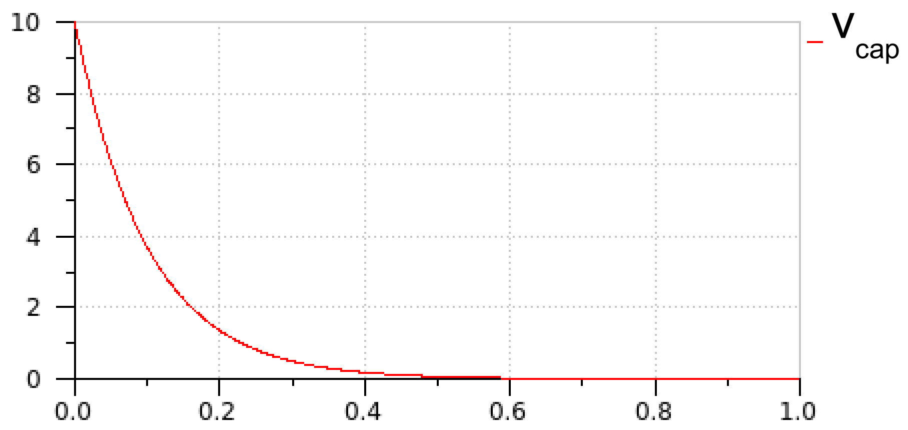
Unrestricted © Siemens AG 2015 All rights reserved.

## Are Block Diagrams Well Suited to Physical Modelling?

From the equations of the model, we can deduce the corresponding block diagram which can be used by a simulator to approximate the actual dynamics by means of numerical simulation:



$$\dot{v}_{cap} = -10 \cdot v_{cap}$$
$$v_{cap}(0) = 10$$

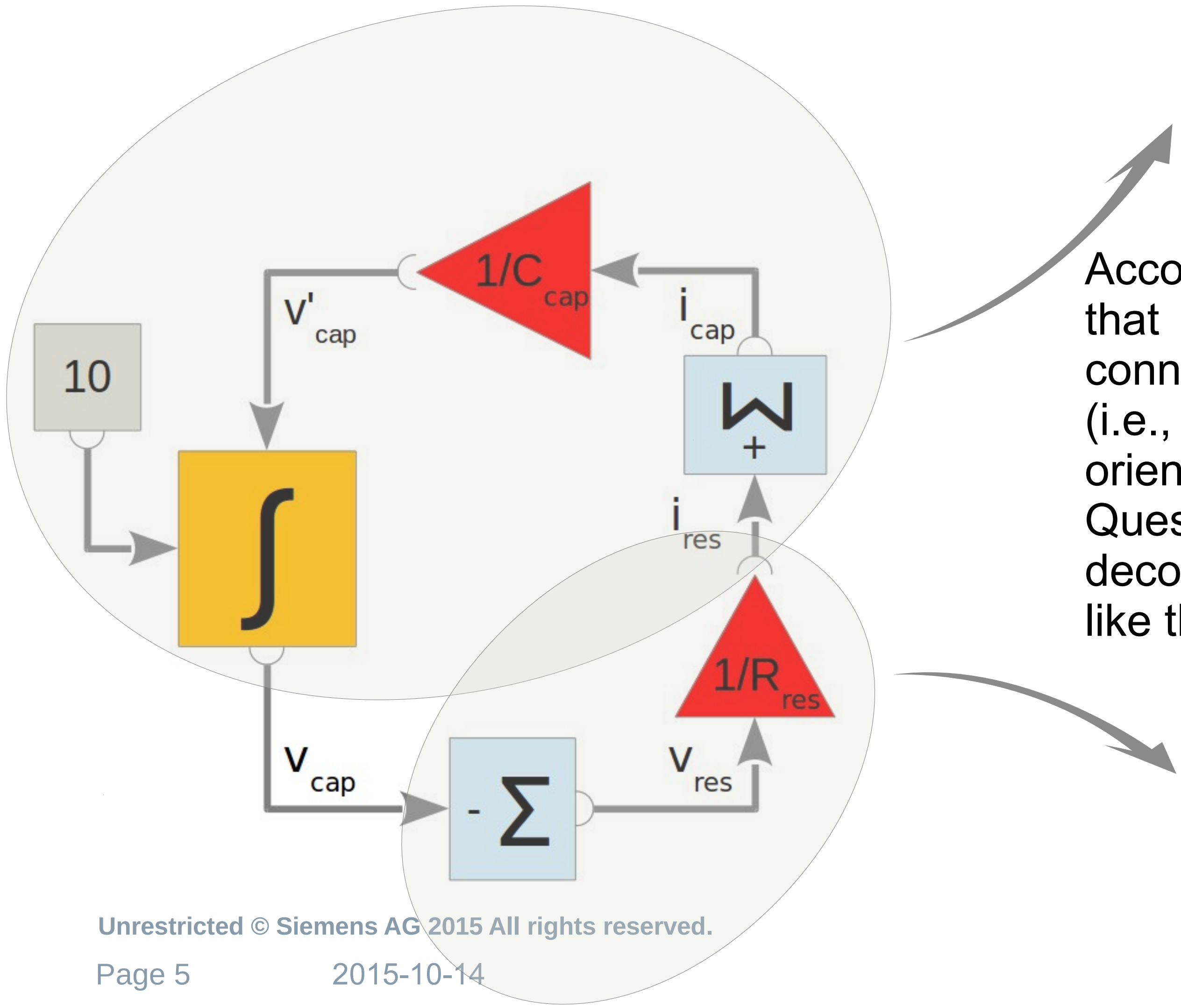


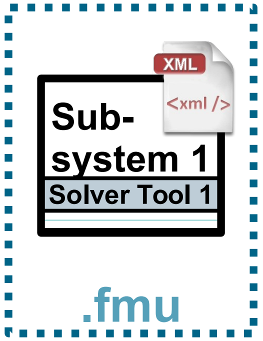
Unrestricted © Siemens AG 2015 All rights reserved.

SIEMENS

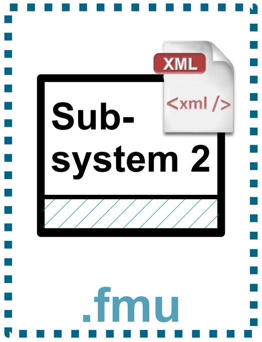
## Are Block Diagrams Well Suited to Physical Modelling?

We can also deduce FMUs from the block diagram with the hope that parts of the original model will be reusable in other contexts





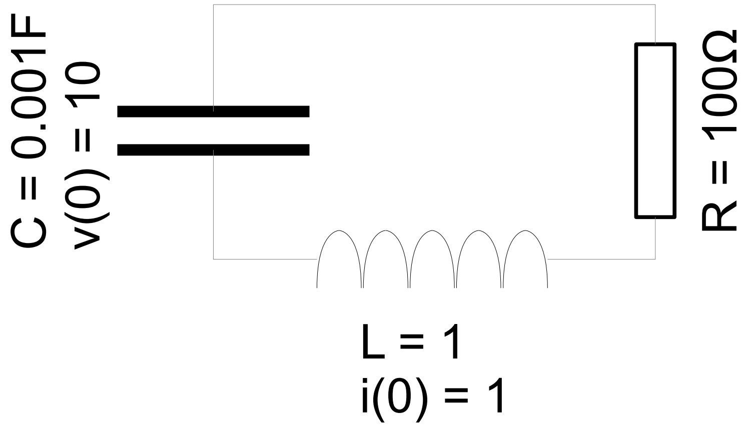
According to this simple example, it seems that building models is as simple as connecting appropriate FMUs together (i.e., corresponding to the correct equation orientation).
Question: is it always possible to decompose a model into simple FMUs like this?



Unrestricted © Siemens AG 2015 All rights reserved.

## Are Block Diagrams Well Suited to Physical Modelling?

Consider this “augmented” version of our original electrical model:



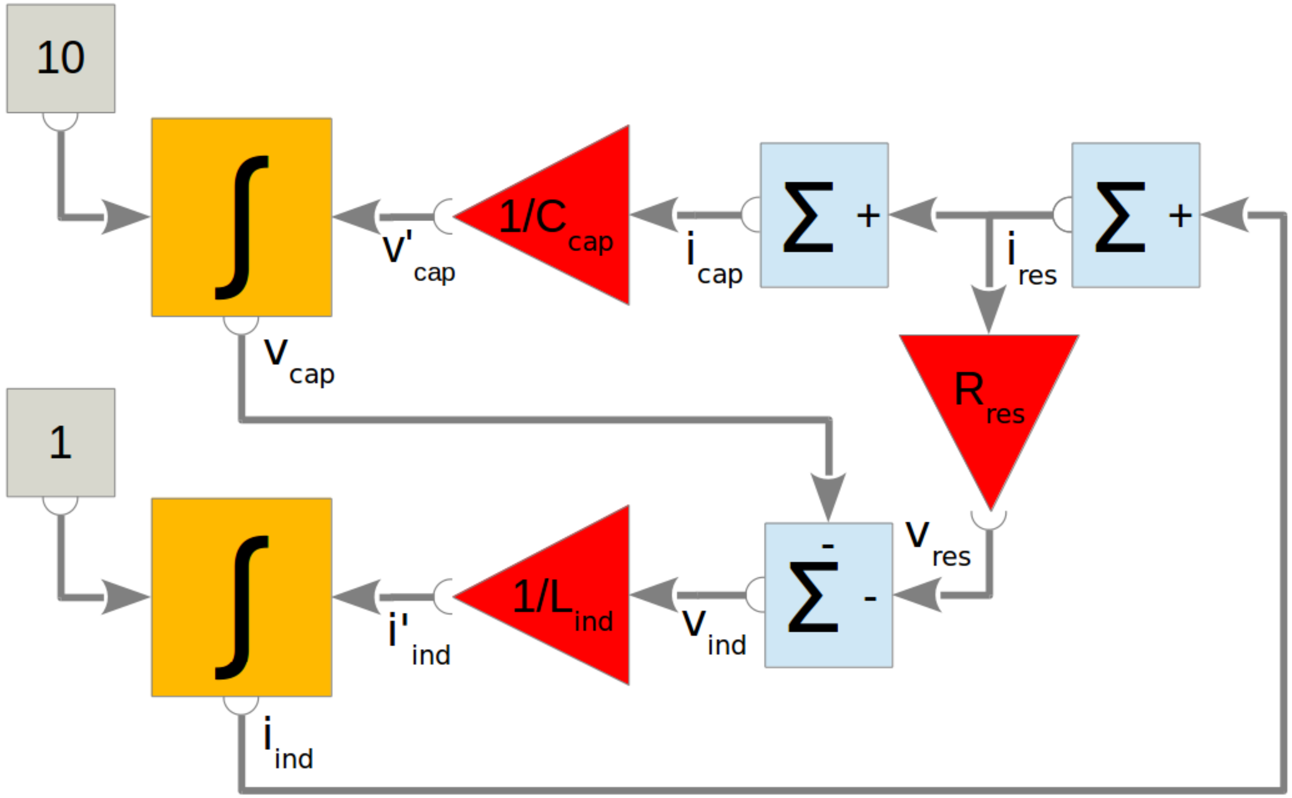
We now have these equations:

$$\begin{cases} C_{cap} \cdot v'_{cap} = i_{cap} & \text{(capacitor's constitutive equation)} \\ v_{res} = R_{res} \cdot i_{res} & \text{(resistor's constitutive equation)} \\ L_{ind} \cdot i'_{ind} = v_{ind} & \text{(inductor's constitutive equation)} \\ i_{cap} = i_{res} & \text{(Kirchhoff's current law)} \\ i_{res} = i_{ind} & \text{(Kirchhoff's current law)} \\ v_{cap} + v_{res} + v_{ind} = 0 & \text{(Kirchhoff's voltage law)} \end{cases}$$

Unrestricted © Siemens AG 2015 All rights reserved.

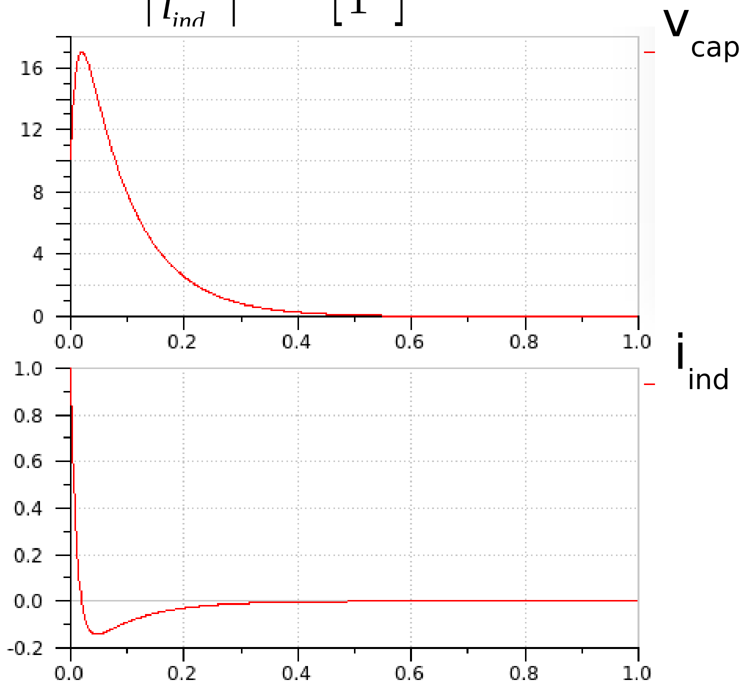
## Are Block Diagrams Well Suited to Physical Modelling?

From the equations of the new model, we obtain the following block diagram representation from which we can also eventually obtain simulation results...



$$\begin{bmatrix} \dot{v}_{cap} \\ \dot{i}_{ind} \end{bmatrix} = \begin{bmatrix} 0 & -1000 \\ -1 & -100 \end{bmatrix} \cdot \begin{bmatrix} v_{cap} \\ i_{ind} \end{bmatrix}$$

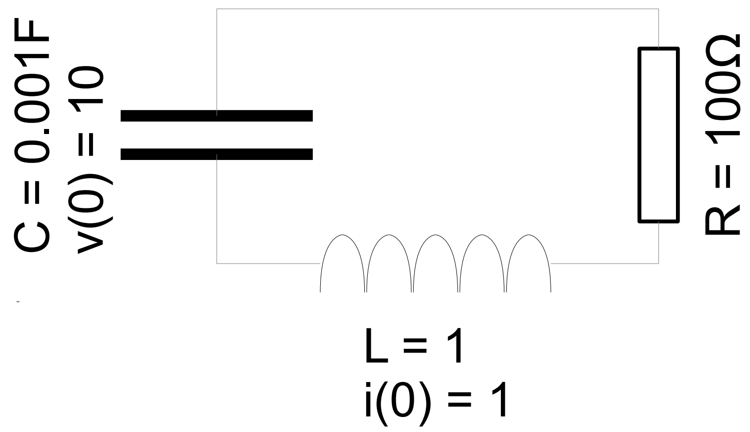
$$\begin{bmatrix} v_{cap} \\ i_{ind} \end{bmatrix}(0) = \begin{bmatrix} 10 \\ 1 \end{bmatrix}$$

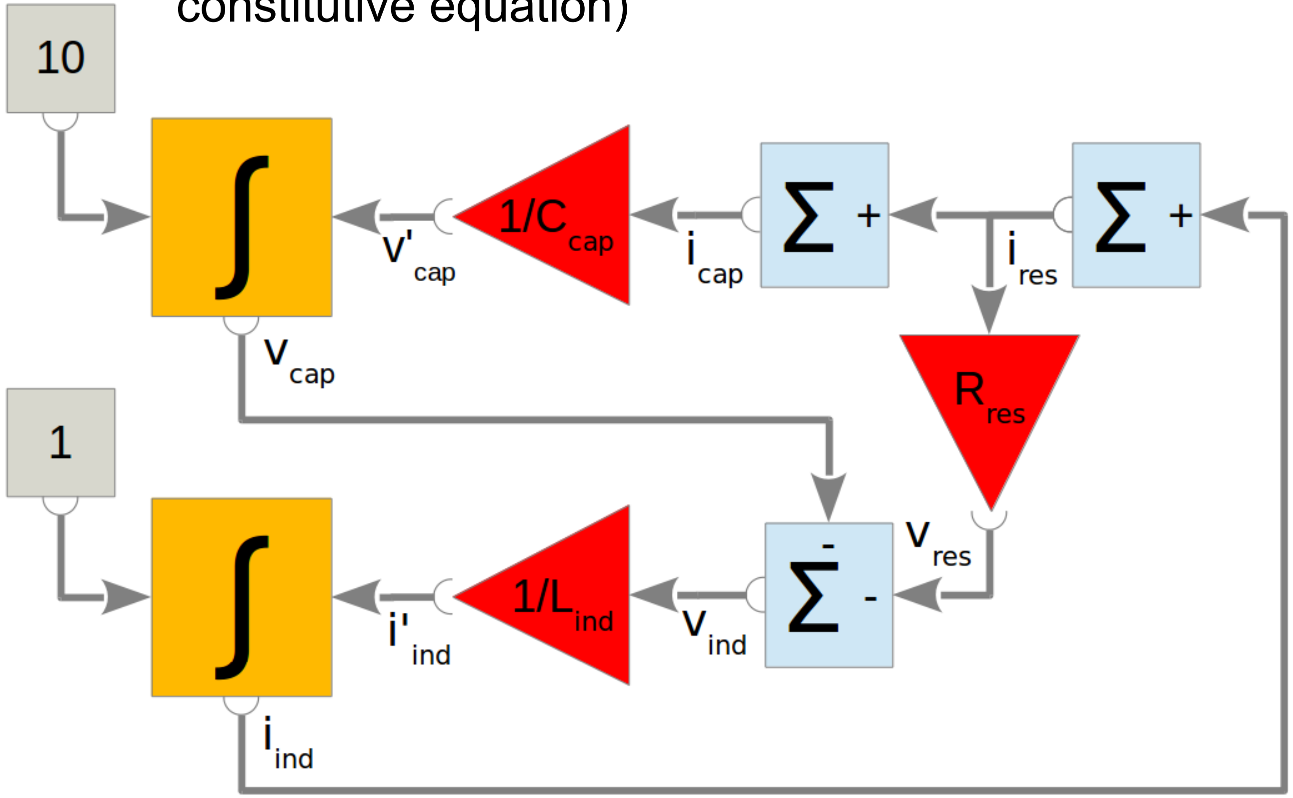


Unrestricted © Siemens AG 2015 All rights reserved.

## Are Block Diagrams Well Suited to Physical Modelling?

… but the topology of the block diagram is no longer comparable with the topology of the original physical model as with the first example (notice also the change of data flow orientation associated with the resistor's constitutive equation)



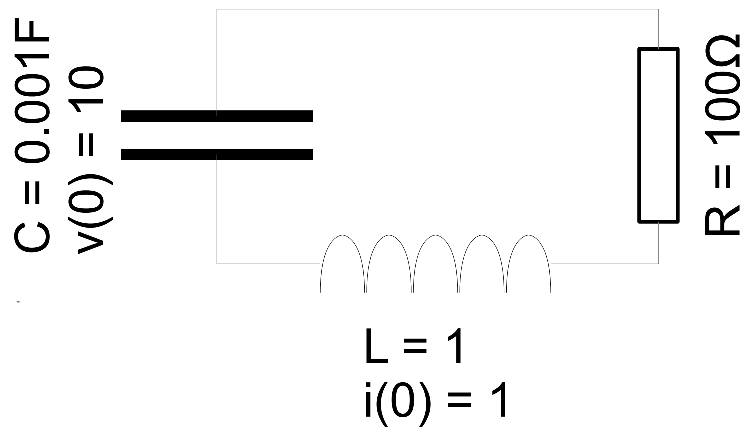


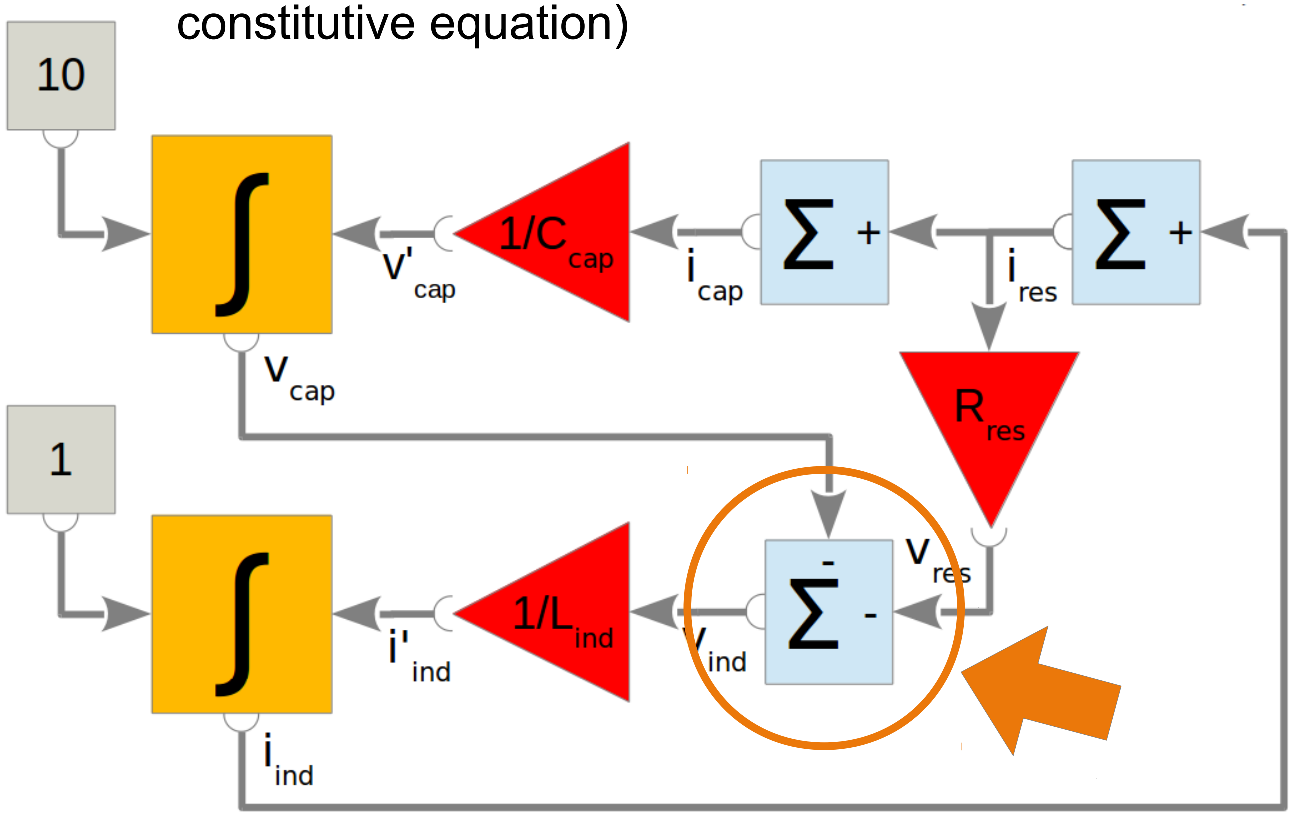
Unrestricted © Siemens AG 2015 All rights reserved.

SIEMENS

## Are Block Diagrams Well Suited to Physical Modelling?

… but the topology of the block diagram is no longer comparable with the topology of the original physical model as with the first example (notice also the change of data flow orientation associated with the resistor's constitutive equation)





In particular, one of the sigma blocks has to be connected to **all** the “physical flows”.

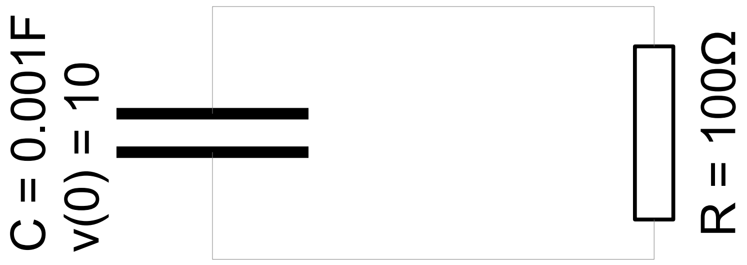
This non-trivial constraint could not have been anticipated in any of the original physical blocks because it results from the **composition operation itself**.

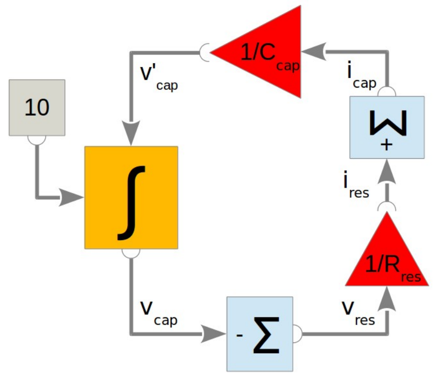
## Are Block Diagrams Well Suited to Physical Modelling?

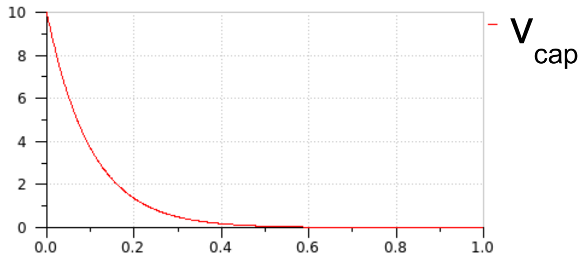
- We have seen that **physical modelling cannot be realistically supported by means of block diagrams**, so new means to connect FMUs should be proposed by the FMI specification in order to support it
- However, **the price to pay for enhancing the FMI proposal with physical modelling facilities (in terms of implementation effort, changes of FMI semantics, etc.) should be compensated with clear benefits (effective modularity, assembly checking capabilities, etc.)**
- Also, **a candidate proposal should ideally preserve current FMI benefits such as the ability to work with model abstractions (no need to reveal the internals of FMUs) and tool neutrality (no need to change the core semantics of FMI compatible tools)**

## Is Modelica Well Suited to Model Based Abstraction?

Recall our very first physical model example:







What happens if we build it using Modelica?

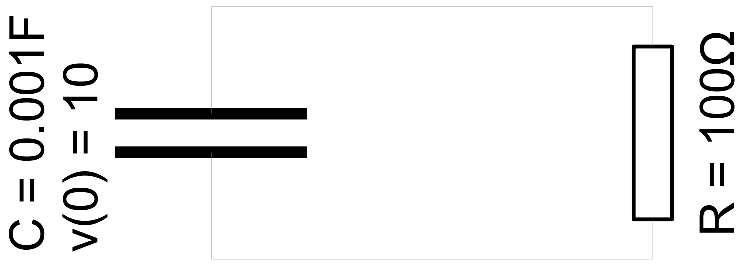
```
model ElectricalExample
  Capacitor
    cap(C = 0.001, v0 = 10);
  Resistor res(R = 100);
equation
  // serial connection
  connect(cap.p, res.n);
  connect(res.p, cap.n);
end ElectricalExample;
```

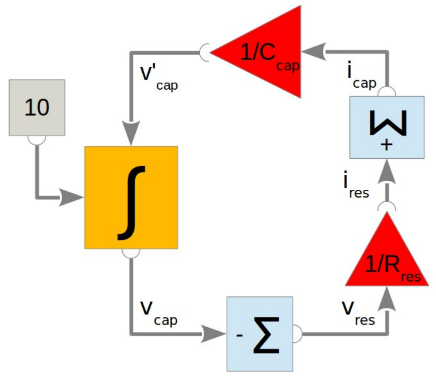
?

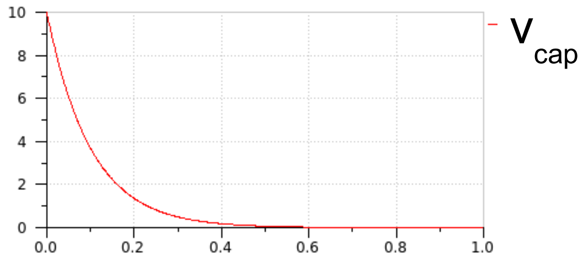
Unrestricted © Siemens AG 2015 All rights reserved.

# Is Modelica Well Suited to Model Based Abstraction?

Recall our very first physical model example:







What happens if we build it using Modelica?

```
model ElectricalExample
  Capacitor
    cap(C = 0.001, v0 = 10);
  Resistor res(R = 100);
equation
  // serial connection
  connect(cap.p, res.n);
  connect(res.p, cap.n);
end ElectricalExample;
```

**Data flow orientation failed.**

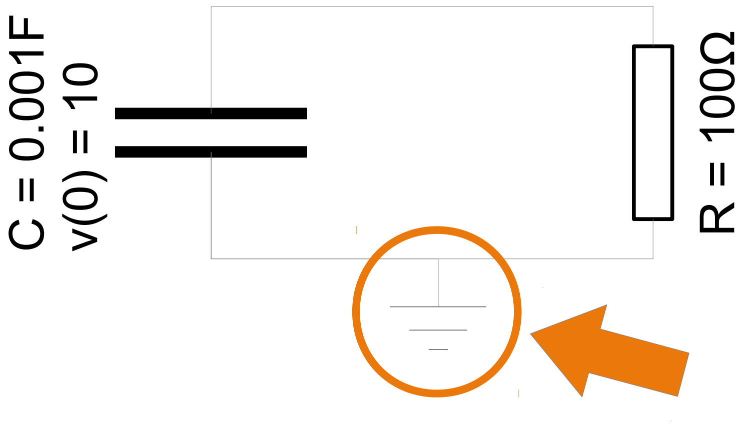
**Structurally singular system having**
**12 equations**
**12 Real variables**

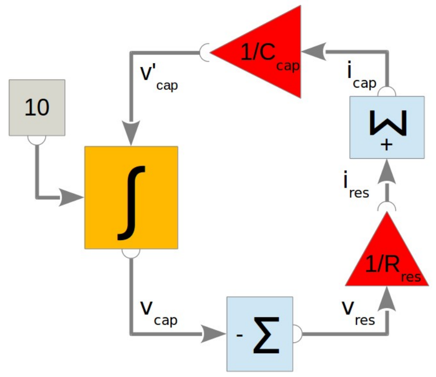
**Missing ground?**

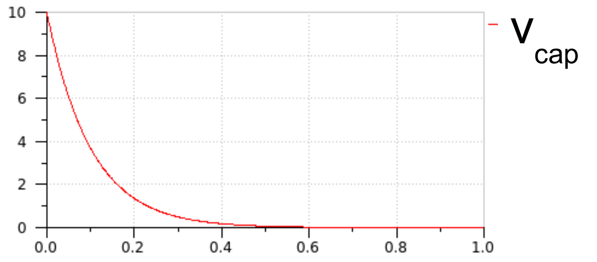
Unrestricted © Siemens AG 2015 All rights reserved.

## Is Modelica Well Suited to Model Based Abstraction?

If we “correct” our initial model by adding a ground, the problem is solved:

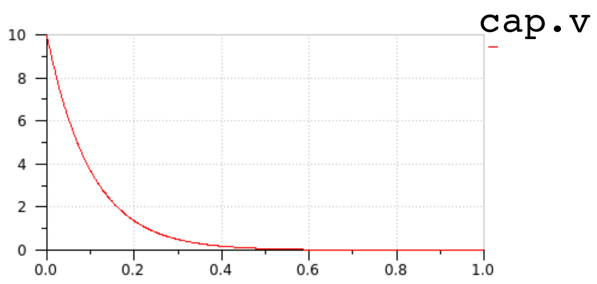






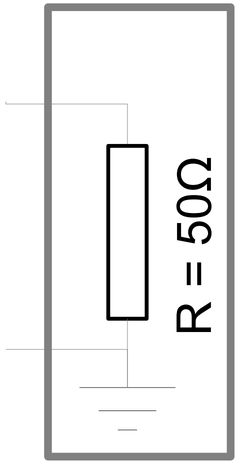
```
model ElectricalExample
  Capacitor
    cap(C = 0.001, v0 = 10);
  Resistor res(R = 100);
  Ground gnd;
equation
  connect(cap.n, gnd.p);
  connect(cap.p, res.n);
  connect(res.p, cap.n);
end ElectricalExample;
```

```
gnd.p.v  := 0;
cap.n.v  := gnd.p.v;
res.n.v  := gnd.p.v;
...
der(cap.v) :=
  cap.i / cap.C;
```



## Is Modelica Well Suited to Model Based Abstraction?

Suppose now that a library designer would have designed the following **grounded** resistor in Modelica:

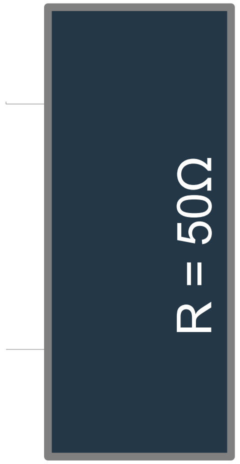


It is important to notice that **in Modelica, equations are used everywhere**: in particular, **the inner structure of the submodel above is unknown from a Modelica compiler**. Modelica definitions are directly transformed into equations which then constitute the only information about the model's semantics.

Unrestricted © Siemens AG 2015 All rights reserved.

## Is Modelica Well Suited to Model Based Abstraction?

In order to protect her IP, the library designer decided to **encrypt** her submodel. Here is the result of applying encryption to the previously unencrypted submodel:

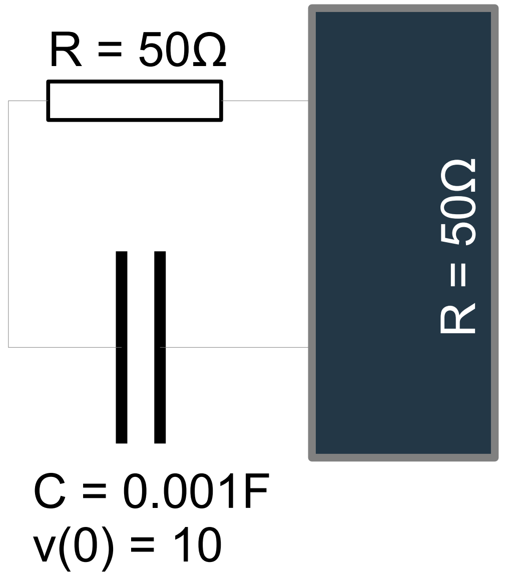


Now, contrary to their favorite Modelica tool which still have access to the model's equations, users no longer have access to any meaningful information about the inners of the submodel (except possibly some variable incidence information).

Unrestricted © Siemens AG 2015 All rights reserved.

## Is Modelica Well Suited to Model Based Abstraction?

Suppose that a user decides to make use of the encrypted submodel defined before to build a simple electrical circuit like this one:

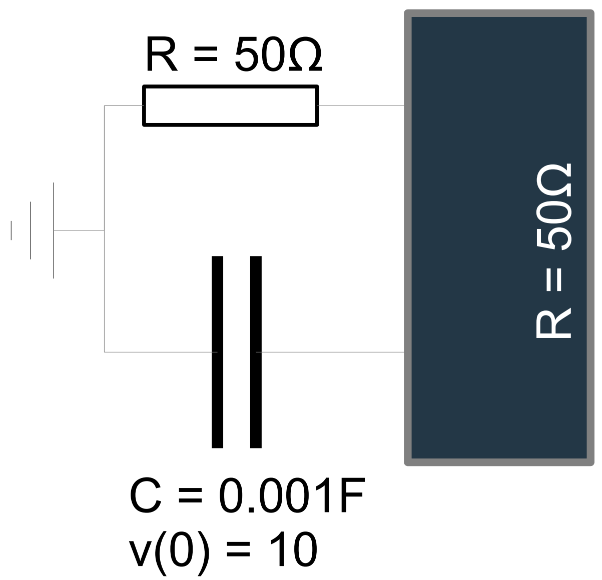


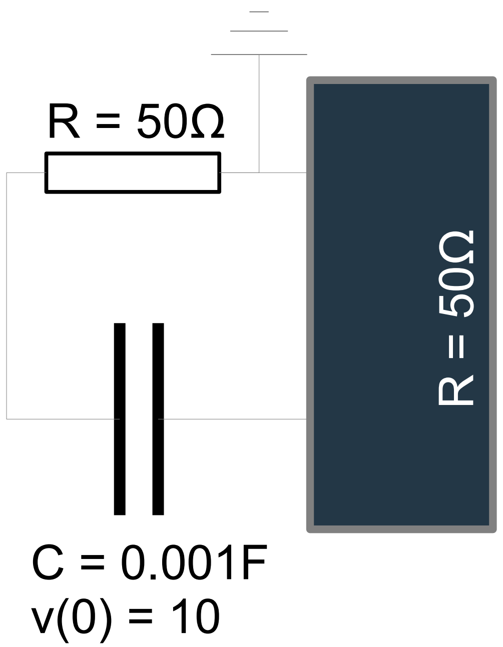
Since it is an experienced Modelica user, he knows that every model has to be grounded. Since he doesn't see any ground in the assembly, he may try to correct his “mistake” before running the simulation...

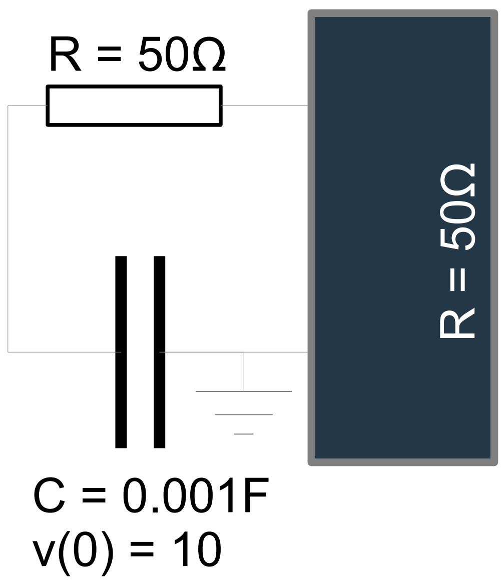
Unrestricted © Siemens AG 2015 All rights reserved.

## Is Modelica Well Suited to Model Based Abstraction?

Here are the three possibilities to add a ground to the model:





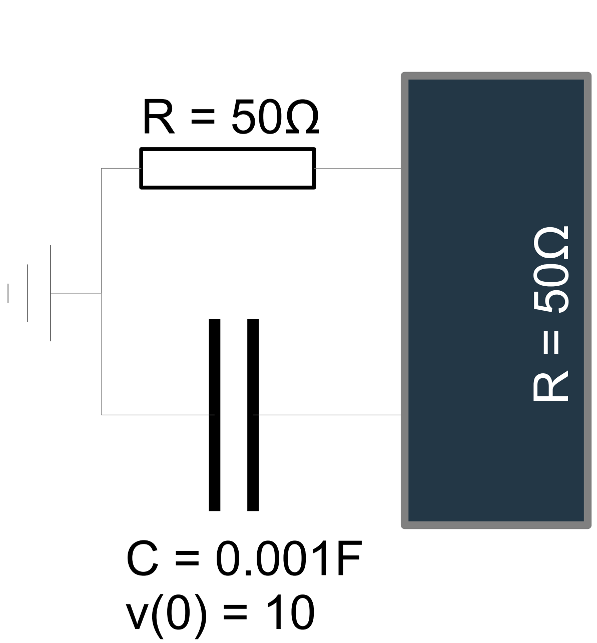


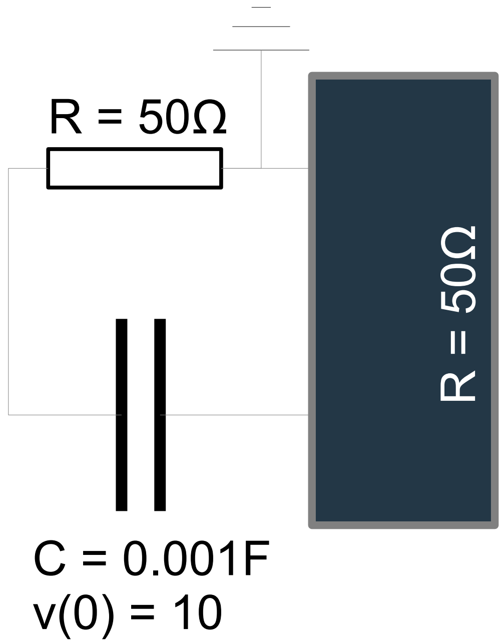
From the user's point of view (remember: he doesn't see innermost grounds), these models seem equivalent: he believes he has just moved the reference point from which to perform voltage measurements (notice that the dynamics of the model should not depend on these measurements).
What happens if he tries to simulate these “corrected” models?

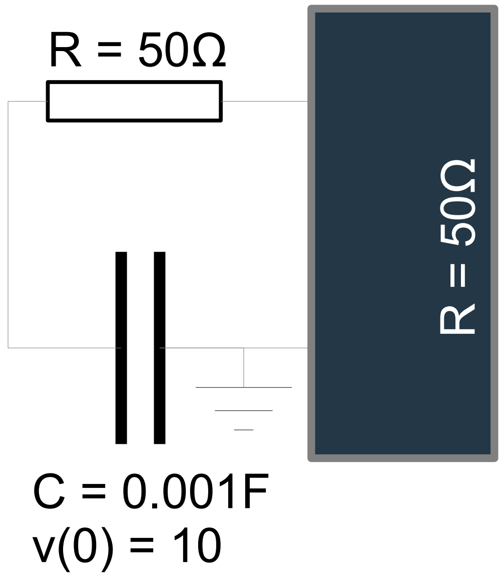
Unrestricted © Siemens AG 2015 All rights reserved.

## Is Modelica Well Suited to Model Based Abstraction?

Unexpectedly, he obtains three different answers! Which one is correct, if any?







**Warning: discarded initial value of**

**cap.v**

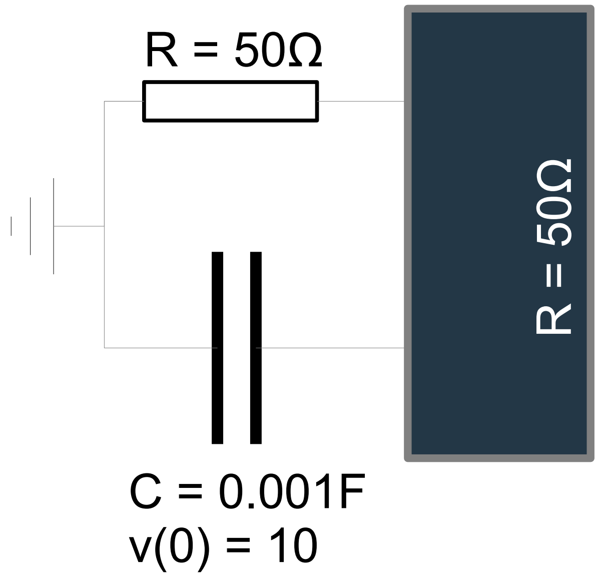
**Simulation successful!**

**Simulation successful!**

**Data flow orientation failed.**
**Structurally singular system having**
**22 equations**
**22 Real variables**

## Is Modelica Well Suited to Model Based Abstraction?

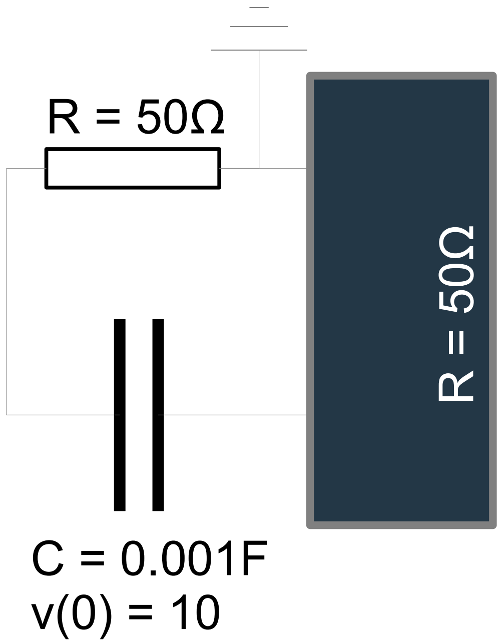
He can already discard the rightmost model since it fails to simulate.
What about, now, the leftmost one, which seems a bit suspicious?



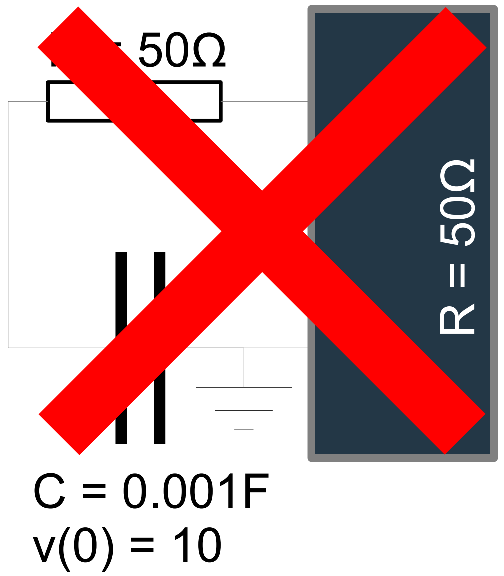
**Warning: discarded initial value of**

**cap.v**

**Simulation successful!**



**Simulation successful!**

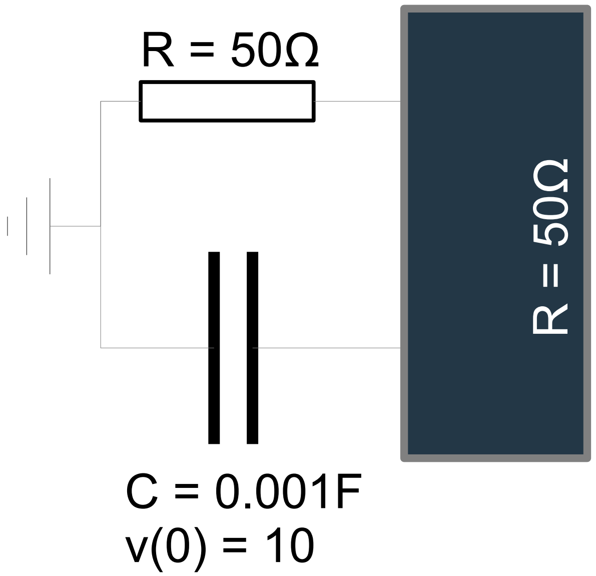


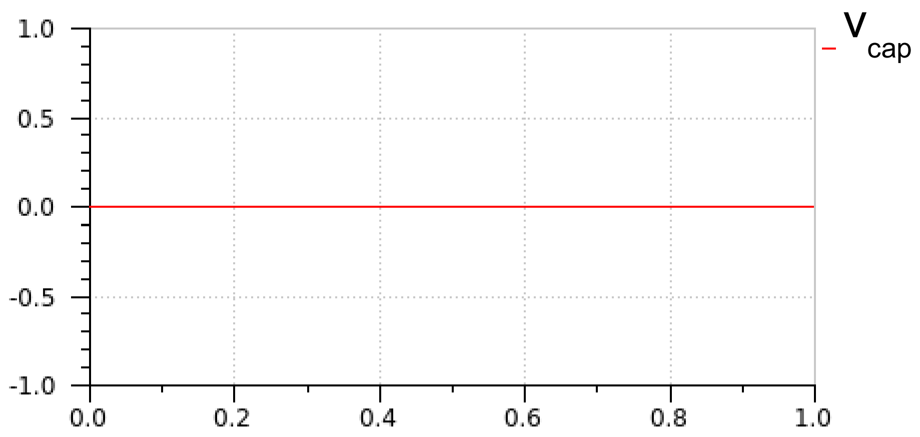
**Data flow orientation failed.**
**Structurally singular system having**
**22 equations**
**22 Real variables**

Unrestricted © Siemens AG 2015 All rights reserved.

## Is Modelica Well Suited to Model Based Abstraction?

Simulation seems to indicate that the model has no dynamics, which is clearly not expected:





**Warning: discarded initial value of**

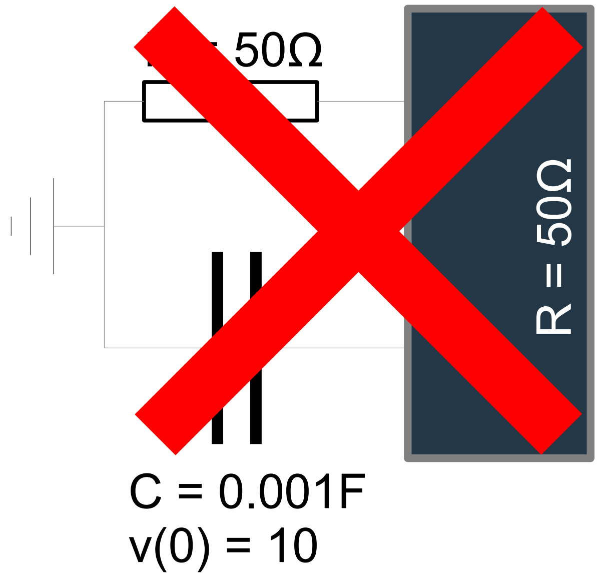
**cap.v**

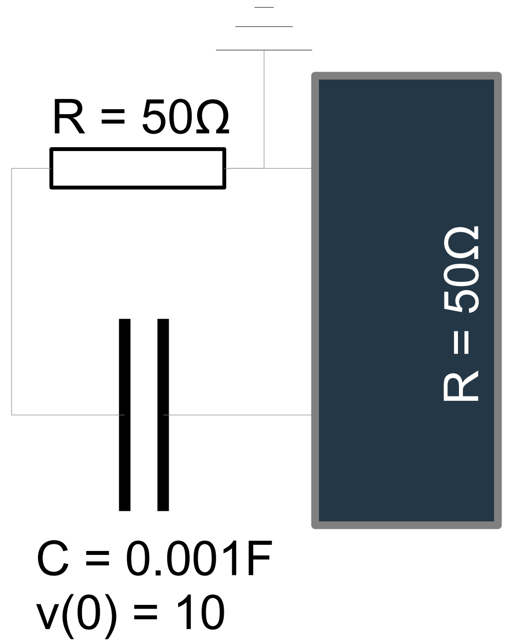
**Simulation successful!**

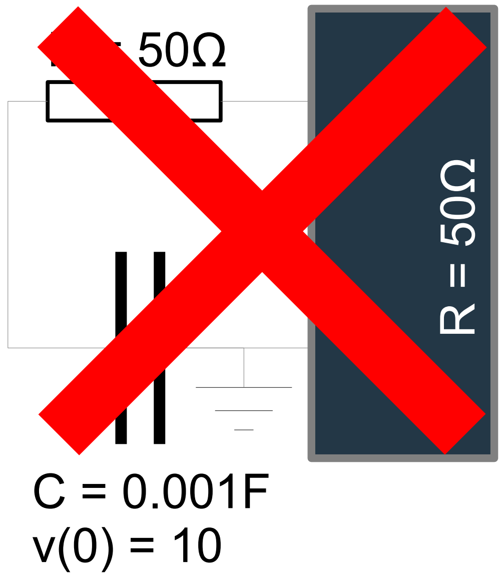
Unrestricted © Siemens AG 2015 All rights reserved.

## Is Modelica Well Suited to Model Based Abstraction?

So the user also discards the leftmost model. What about the one in the middle?







**Warning: discarded initial value of**

**cap.v**

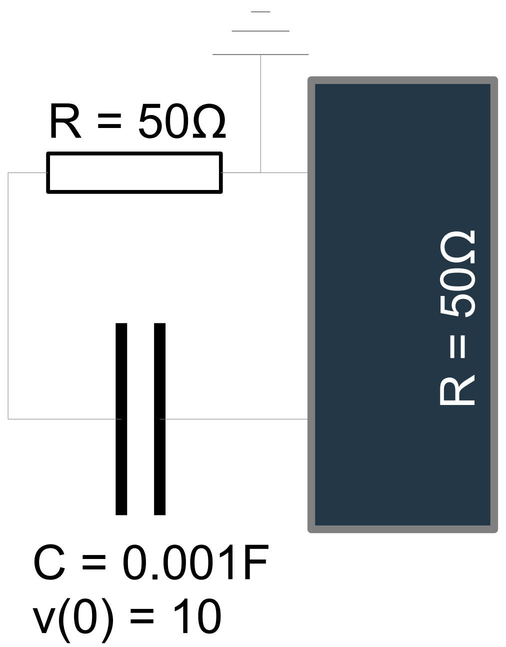
**Simulation successful!**

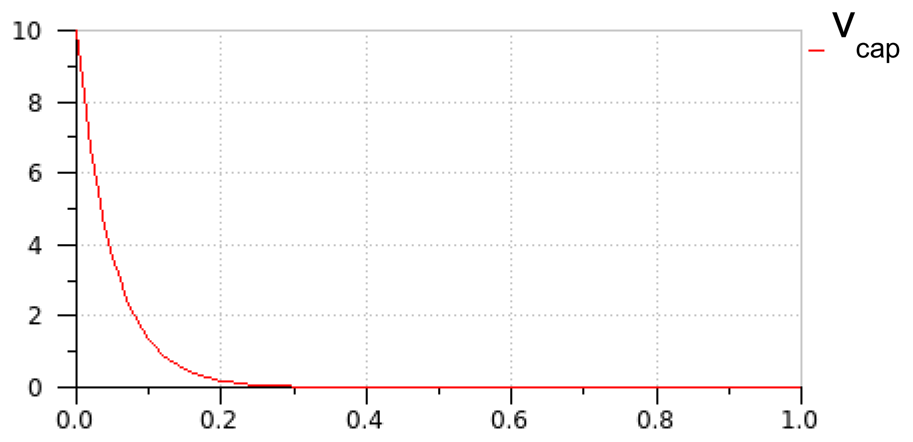
**Simulation successful!**

**Data flow orientation failed.**
**Structurally singular system having**
**22 equations**
**22 Real variables**

## Is Modelica Well Suited to Model Based Abstraction?

Simulation seems correct since one can observe the typical exponential decay of an RC circuit:



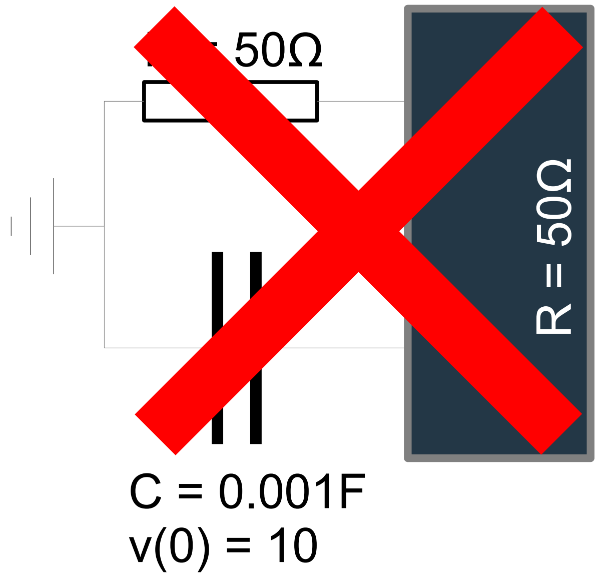


**Simulation successful!**

Unrestricted © Siemens AG 2015 All rights reserved.

## Is Modelica Well Suited to Model Based Abstraction?

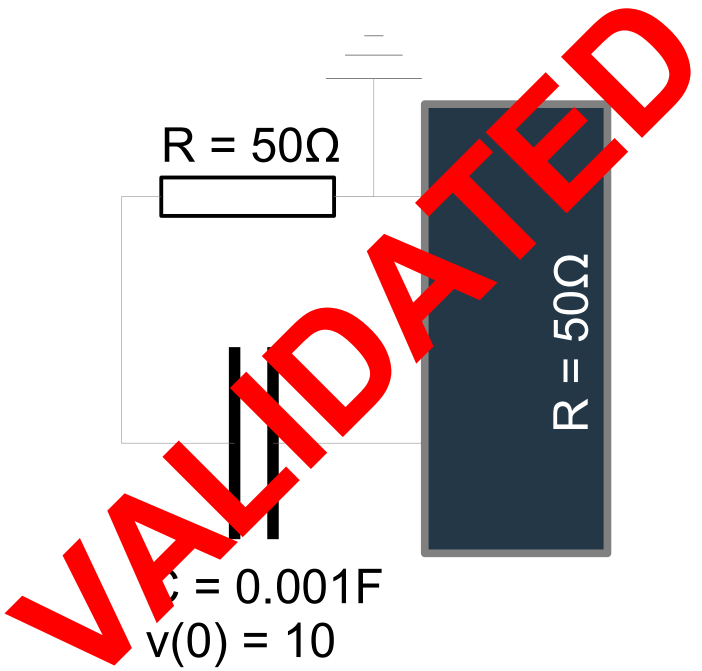
So the user is tempted to validate this model...



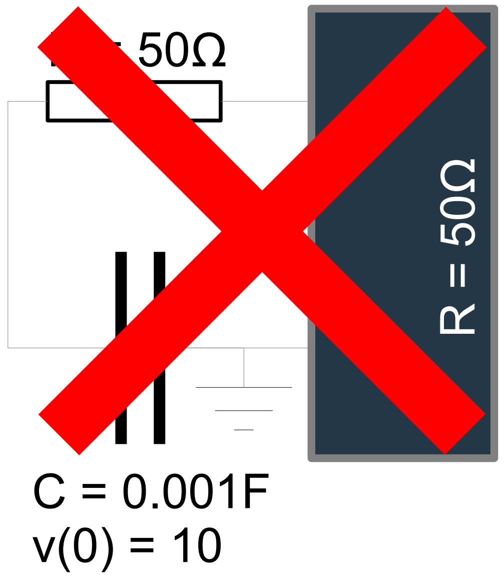
**Warning: discarded initial value of**

**capacitor.v**

**Simulation successful!**



**Simulation successful!**

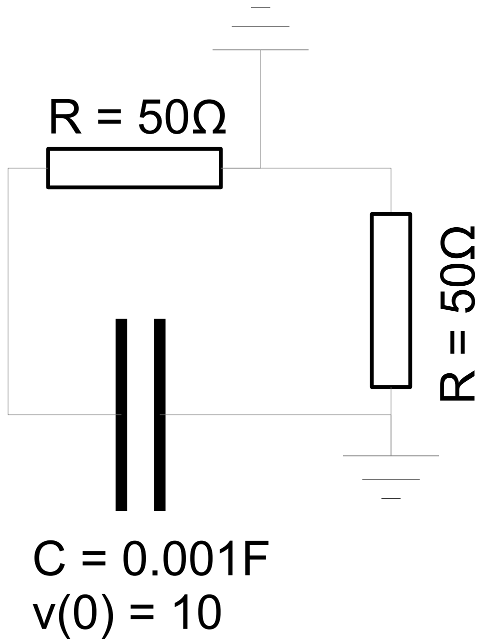


**Data flow orientation failed.**
**Structurally singular system having**
**22 equations**
**22 Real variables**

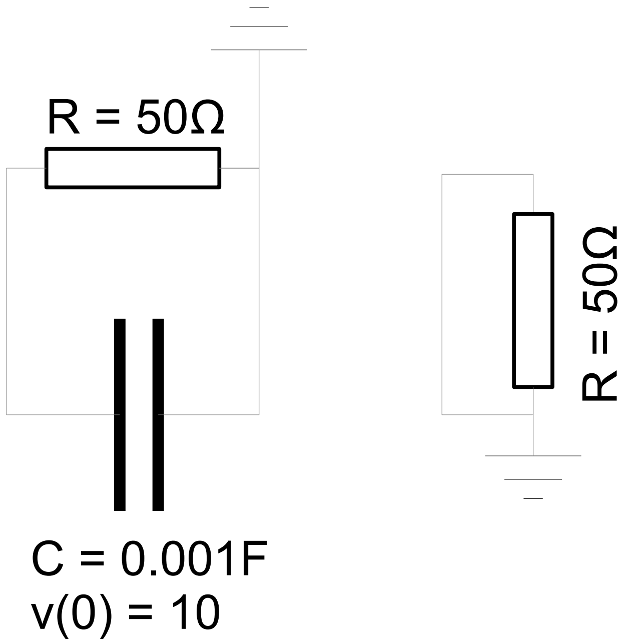
Unrestricted © Siemens AG 2015 All rights reserved.

## Is Modelica Well Suited to Model Based Abstraction?

… But if he would have had a way to recover the **initial structure** of the encrypted submodel, this would have revealed that **the "validated" model was probably not the one he was having in mind**:



is actually equivalent to:

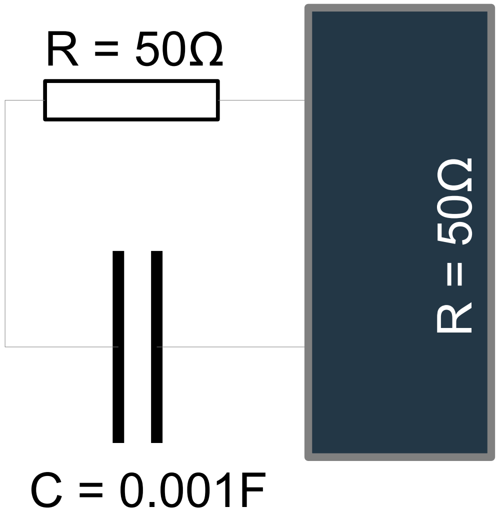


**Because of the lack of structural information in Modelica, the user had no way to figure out *a priori* that no power exchange would actually occur between the grounded resistor and the rest of the model!**

Unrestricted © Siemens AG 2015 All rights reserved.

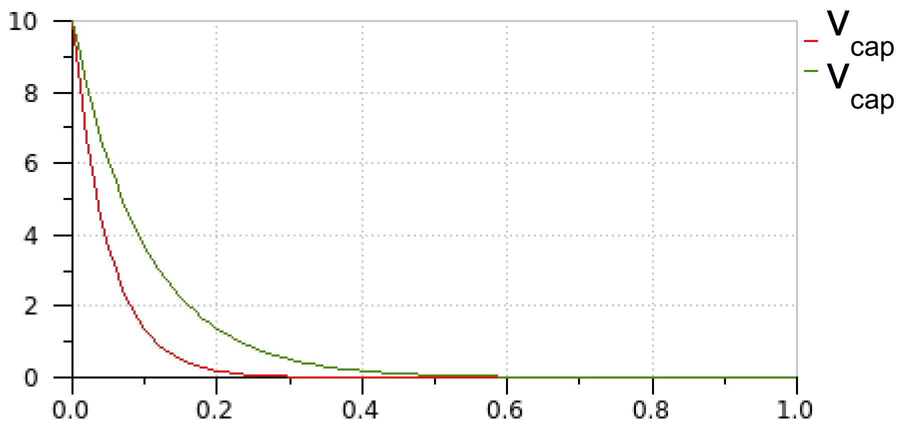
## Is Modelica Well Suited to Model Based Abstraction?

Actually, the user should not have had added any ground to the model to get the correct, expected result:



v(0) = 10

**Simulation successful!**



In green: the correct result
In red: result obtained with the "validated" model

**This raises the following questions. How to make sure in practice:**

- **that a model is simply complete (!),**
- **that power exchanges actually occur,**
- **that a simulation failure is not caused by a source conflict,**
- **etc.**

Unrestricted © Siemens AG 2015 All rights reserved.

## Is Modelica Well Suited to Model Based Abstraction?

- **Many kinds of structural errors may occur when connecting Modelica physical models** (e.g., conflicting sources, constrained state variables, missing connections, etc.), **some of them even leading to simulation results, but wrong ones**
- Also, most of the time, **Modelica compilers issue poor error messages**
  - **Even in case the source code of the FMU is given, errors are still explained in terms of "missing equations", "over-constrained subsystems", "singular incidence matrices", etc., which are clearly not the right level of explanation expected by users of physical models** (notice moreover that **users and authors of models are generally not the same persons**)
  - **The two alternatives offered by Modelica are: many equations with some compiler "hints", and IP protection with at best some obscure "incidence information"**

**Unrestricted © Siemens AG 2015 All rights reserved.**

## Proposal

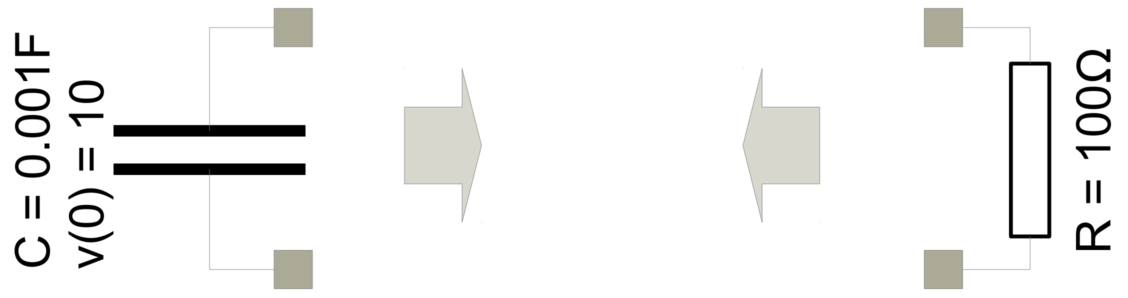
## Graph Structure of Lumped Parameter Models of Systems

- It can be shown that **finite, lumped parameter models can always be described by means of oriented linear graphs** whose arcs represent **implicit** relations involving so-called **across** and **through** variables (**Horace Trent, Isomorphisms between Oriented Linear Graphs and Lumped Physical Systems, 1955**)
- Moreover, **connection equations can be obtained directly from the graph structure** (without any peek at implicit relations) by means of classical graph algorithms (cycle and co-cycle base determination)
- So it seems that we have some ingredients to design an **approach where graph representation of models and of their types play a central role**...

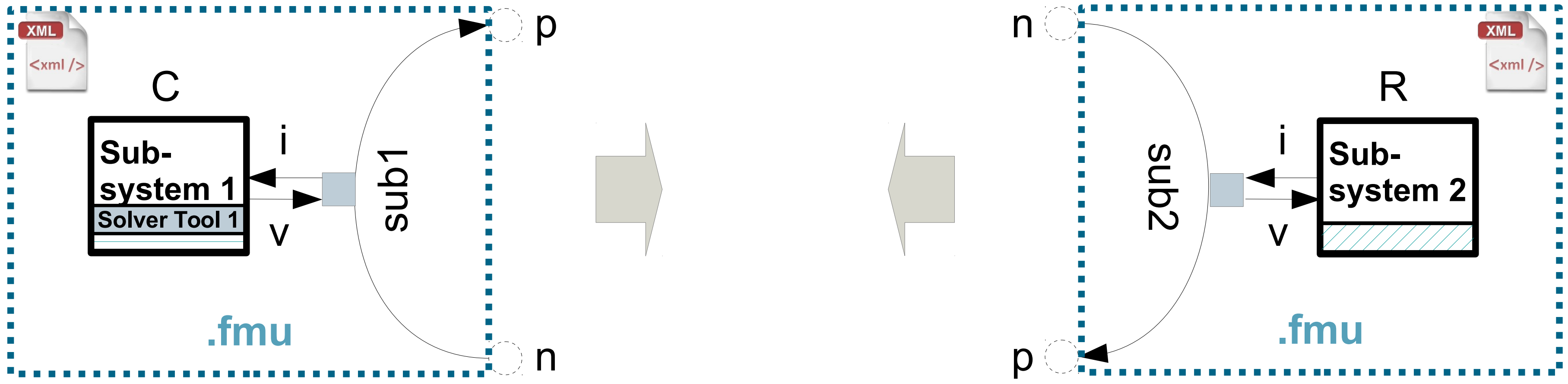
**Unrestricted © Siemens AG 2015 All rights reserved.**

## Application of Linear Graphs to “FMI with physical ports”

Let's illustrate the idea with our very first simple example:



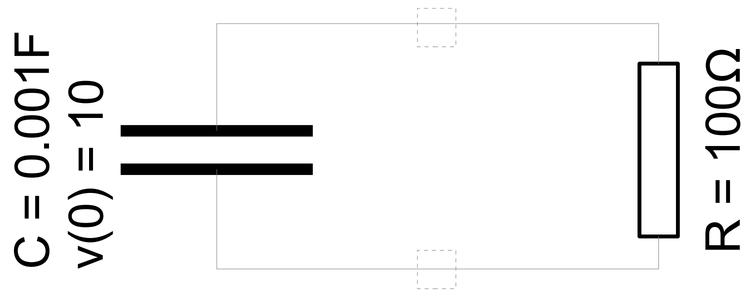
The key idea is to **not expose directly physical ports**, but **nodes—which do not expose any variables—of an explicit linear graph structure holding “classical” FMUs** (i.e. FMUs with oriented data flow):



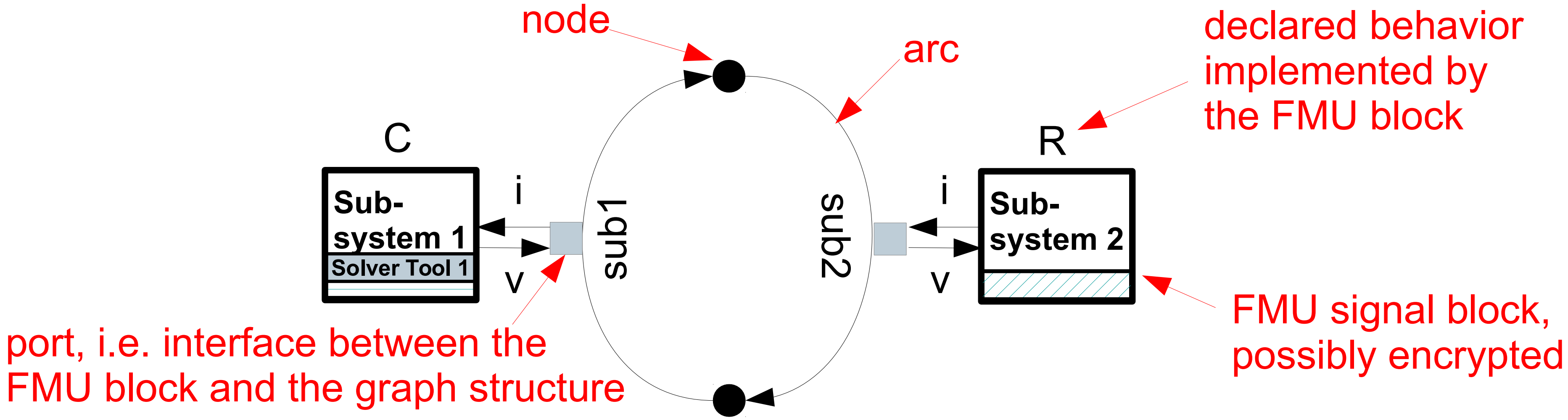
Unrestricted © Siemens AG 2015 All rights reserved.

## Application of Linear Graphs to “FMI with physical ports”

Our original model was composed of interconnected C and R elements:



The corresponding FMU assembly is composed of an **explicit graph structure** made of two interconnected arcs holding **signal based, possibly encrypted FMUs**:

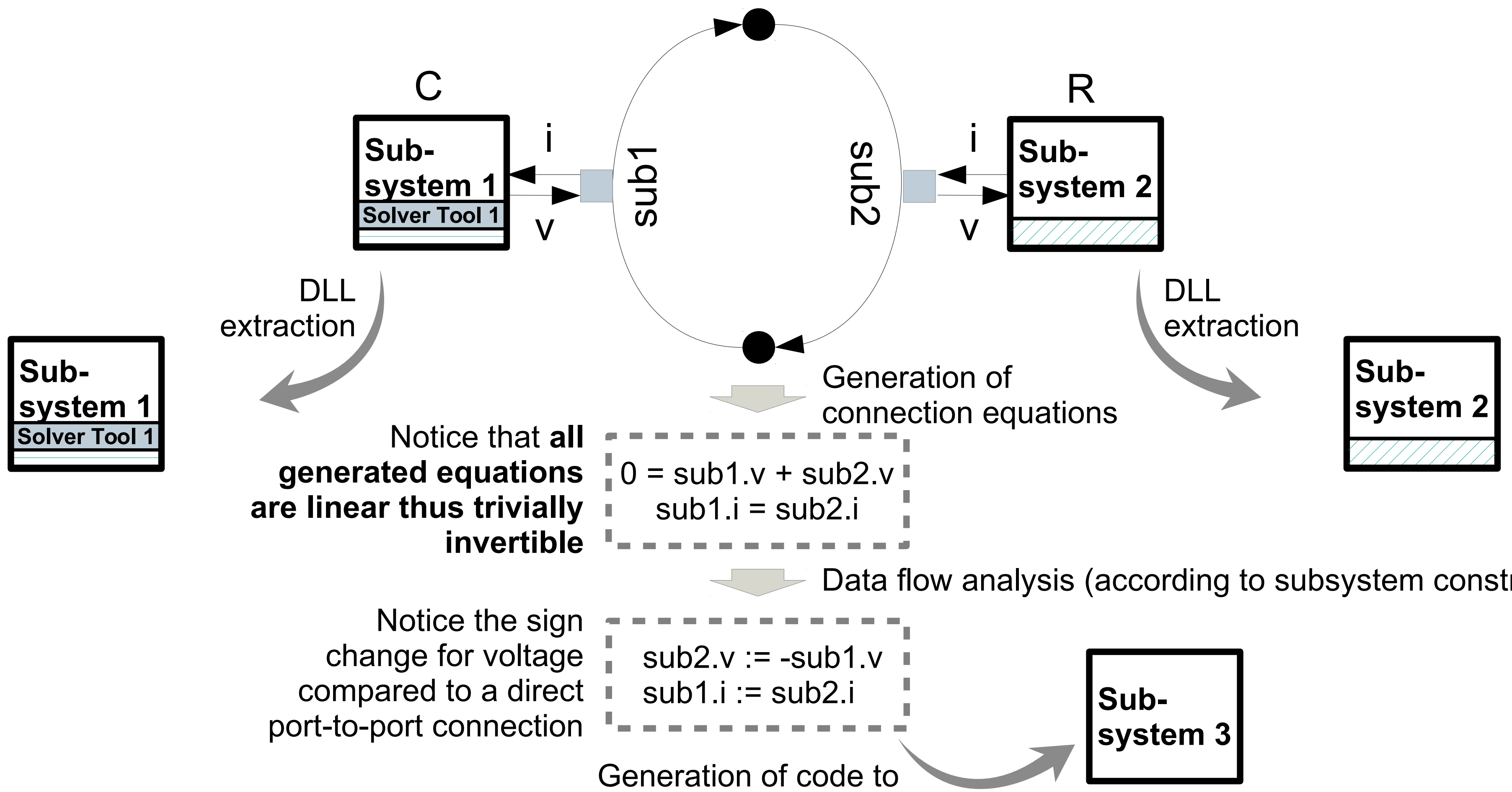

node
arc
declared behavior implemented by the FMU block
C
R
Sub-system 1
Solver Tool 1
Sub-system 2
i
v
sub1
sub2
port, i.e. interface between the FMU block and the graph structure
FMU signal block, possibly encrypted

Unrestricted © Siemens AG 2015 All rights reserved.

SIEMENS

## Application of Linear Graphs to “FMI with physical ports”

It is now possible generate **connection equations** by exploiting the graph structure and orientation constraints attached to FMUs:



Generation of connection equations

Notice that **all generated equations are linear thus trivially invertible**

0 = sub1.v + sub2.v
sub1.i = sub2.i

Data flow analysis (according to subsystem constraints)

Notice the sign change for voltage compared to a direct port-to-port connection

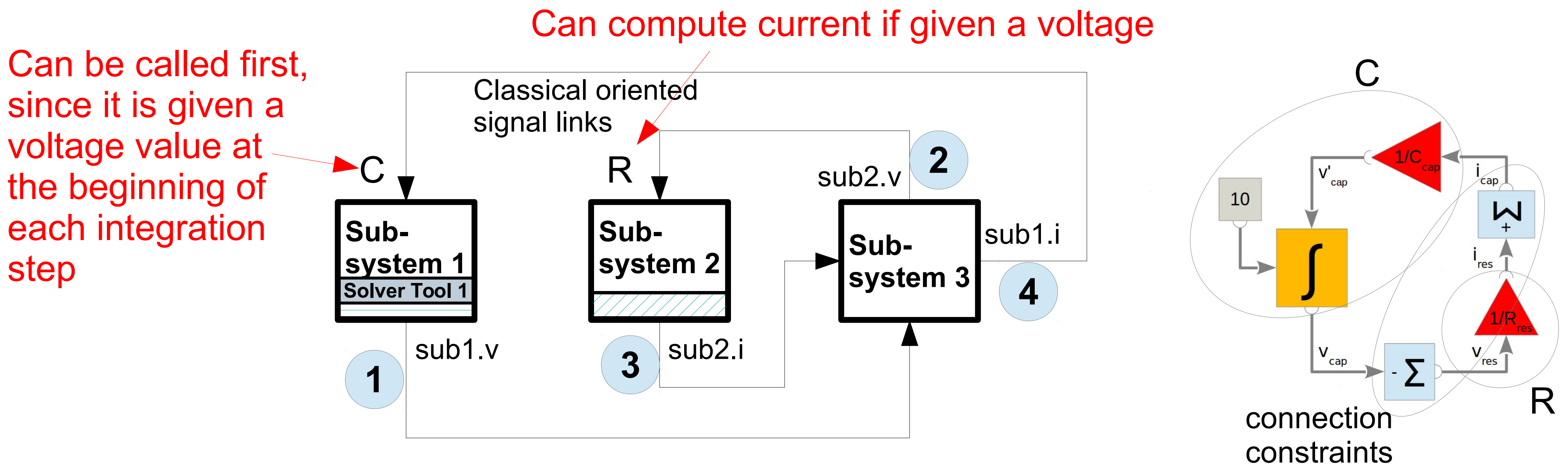
sub2.v := -sub1.v
sub1.i := sub2.i

Generation of code to handle oriented connection constraints (for example as a subsystem)

Unrestricted © Siemens AG 2015 All rights reserved.

# Application of Linear Graphs to “FMI with physical ports”

Last, it is possible to determine the **oriented data flow of the “executable” model**:


Can compute current if given a voltage
Can be called first, since it is given a voltage value at the beginning of each integration step
Classical oriented signal links
C
R
Sub-system 1
Solver Tool 1
Sub-system 2
Sub-system 3
sub2.v
sub1.i
sub1.v
sub2.i
1
2
3
4
C
10
1/C cap
v' cap
i cap
i res
1/R res
v cap
v res
Σ
R
connection constraints

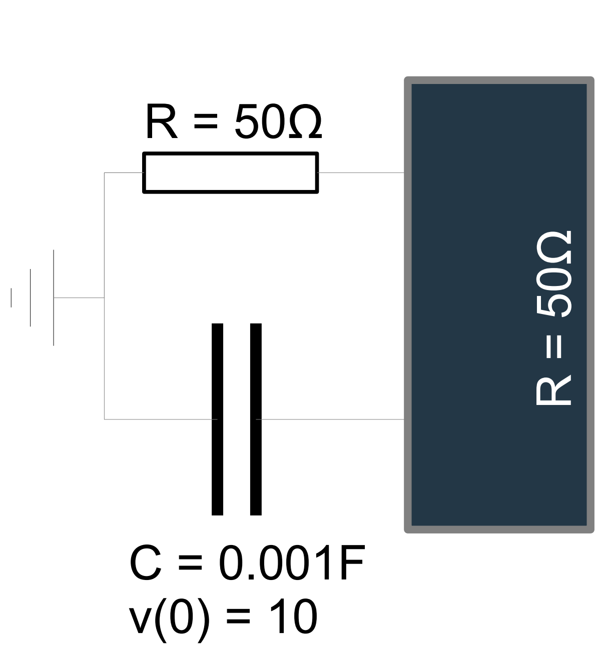
**Notice that we never have had to “open” FMU signal blocks in order to determine the correct calling sequence, neither have we had to manipulate any equation other than the linear ones generated by means of the graph based algorithms**

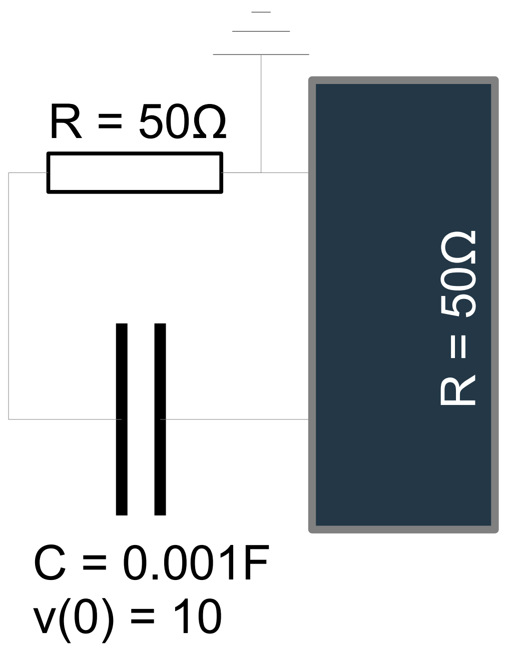
Unrestricted © Siemens AG 2015 All rights reserved.

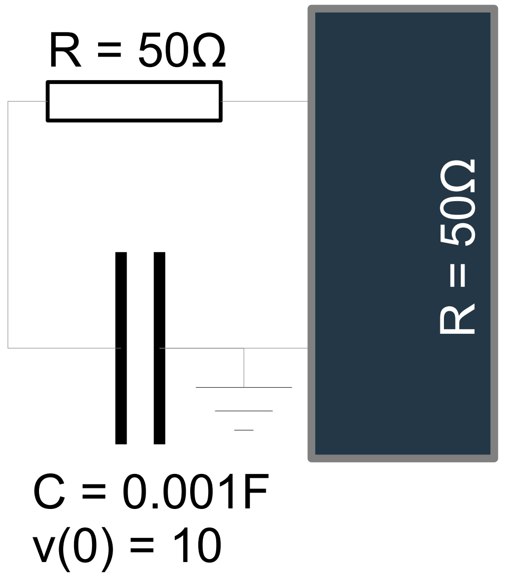
SIEMENS

## Application of Linear Graphs to “FMI with physical ports”

What about our three “pathological” models defined previously?



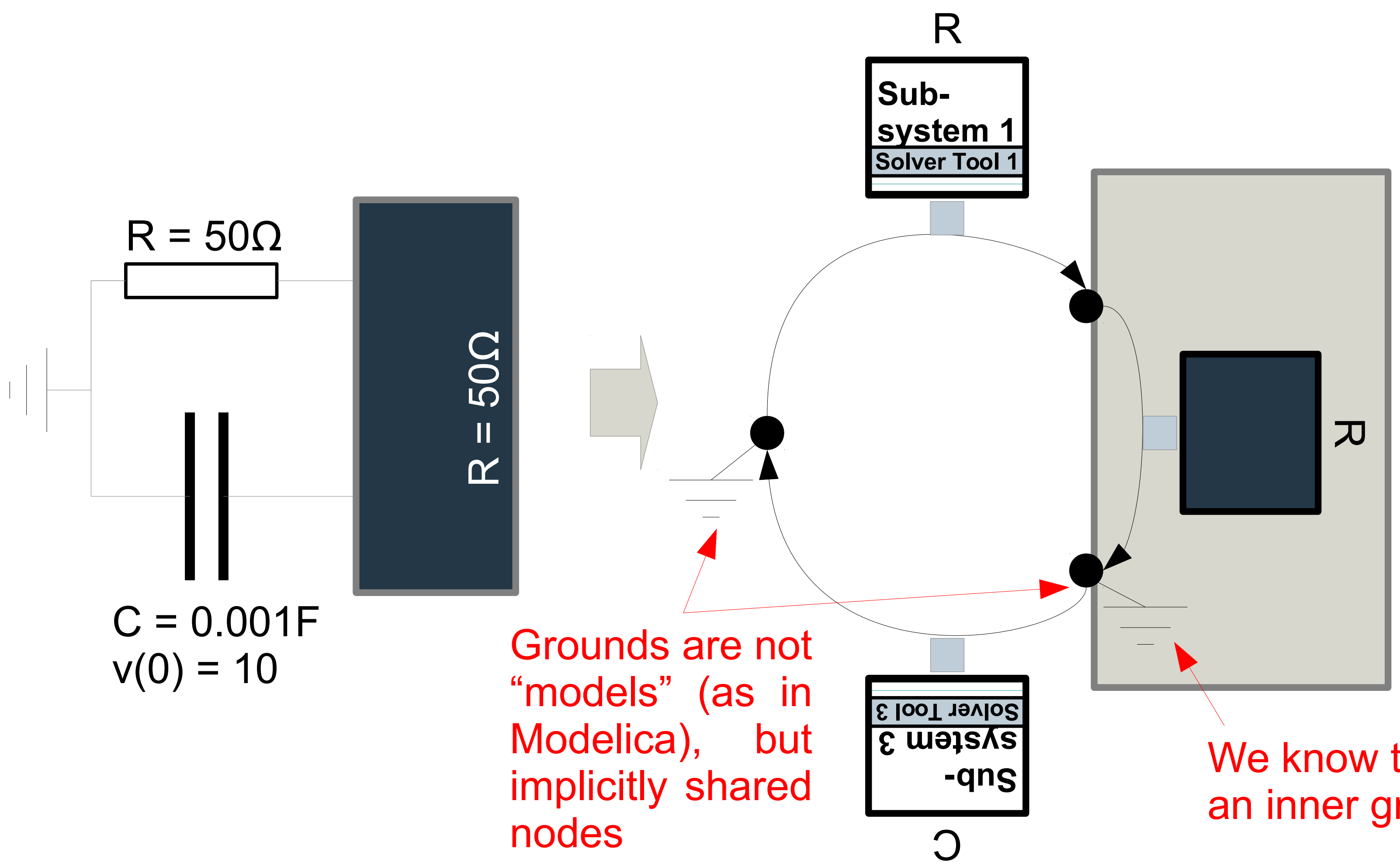




Unrestricted © Siemens AG 2015 All rights reserved.

## Application of Linear Graphs to "FMI with physical ports"

The first model can be defined like this:



Grounds are not "models" (as in Modelica), but implicitly shared nodes

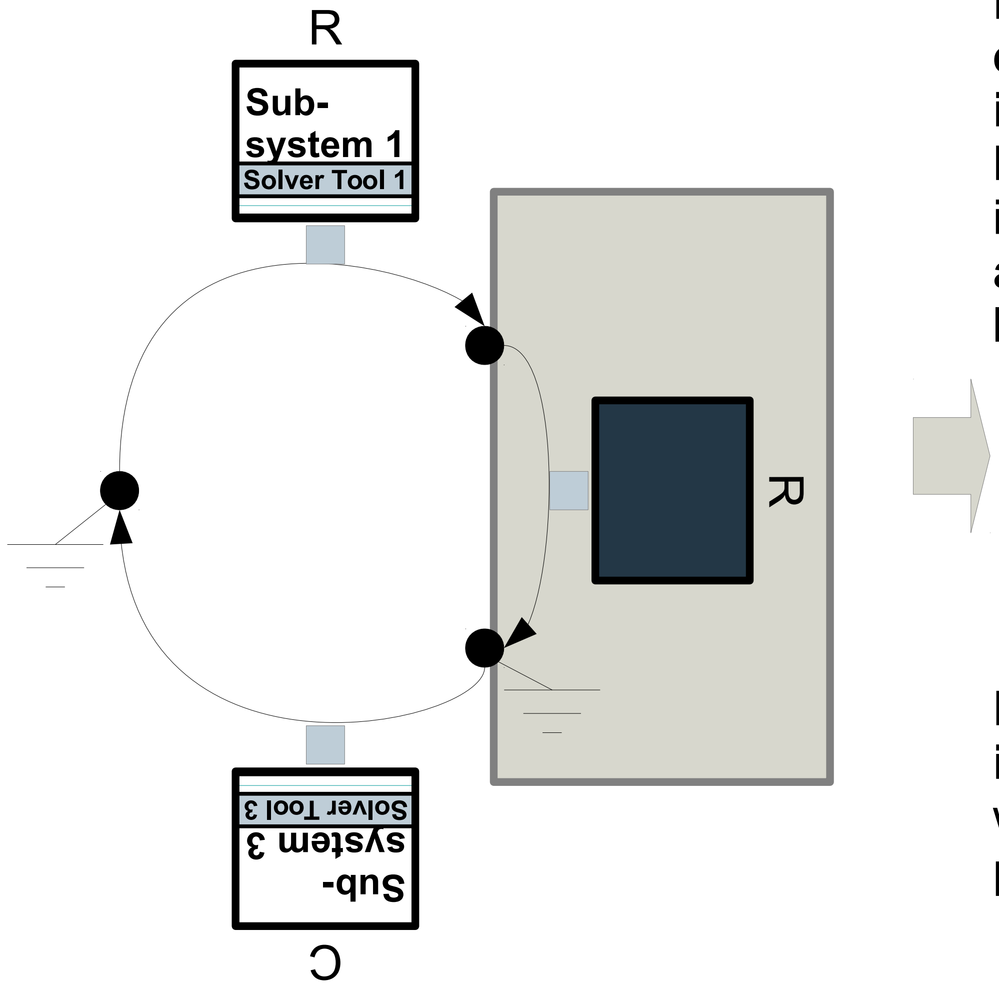
The "physical FMU" is translucent: the graph structure is visible, but not the inners of the FMU signal block which performs the "interesting" calculations

We know that we have an inner ground

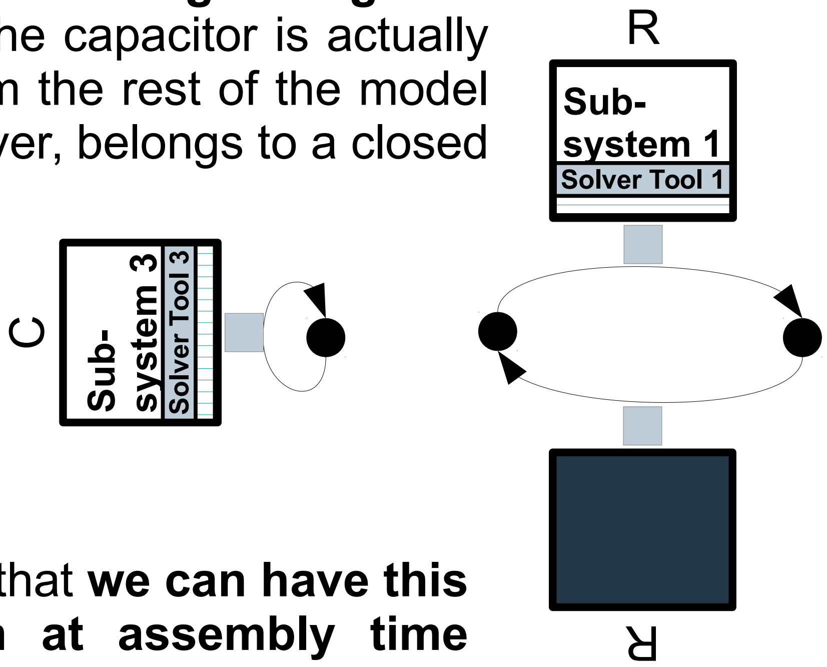
Unrestricted © Siemens AG 2015 All rights reserved.

## Application of Linear Graphs to “FMI with physical ports”

After removal of grey boxes and graphical resolution of implicitly shared nodes constraints, we finally get:



Notice that **we now have an explanation for the mysterious initial value message we got in Modelica**: the capacitor is actually isolated from the rest of the model and, moreover, belongs to a closed loop.

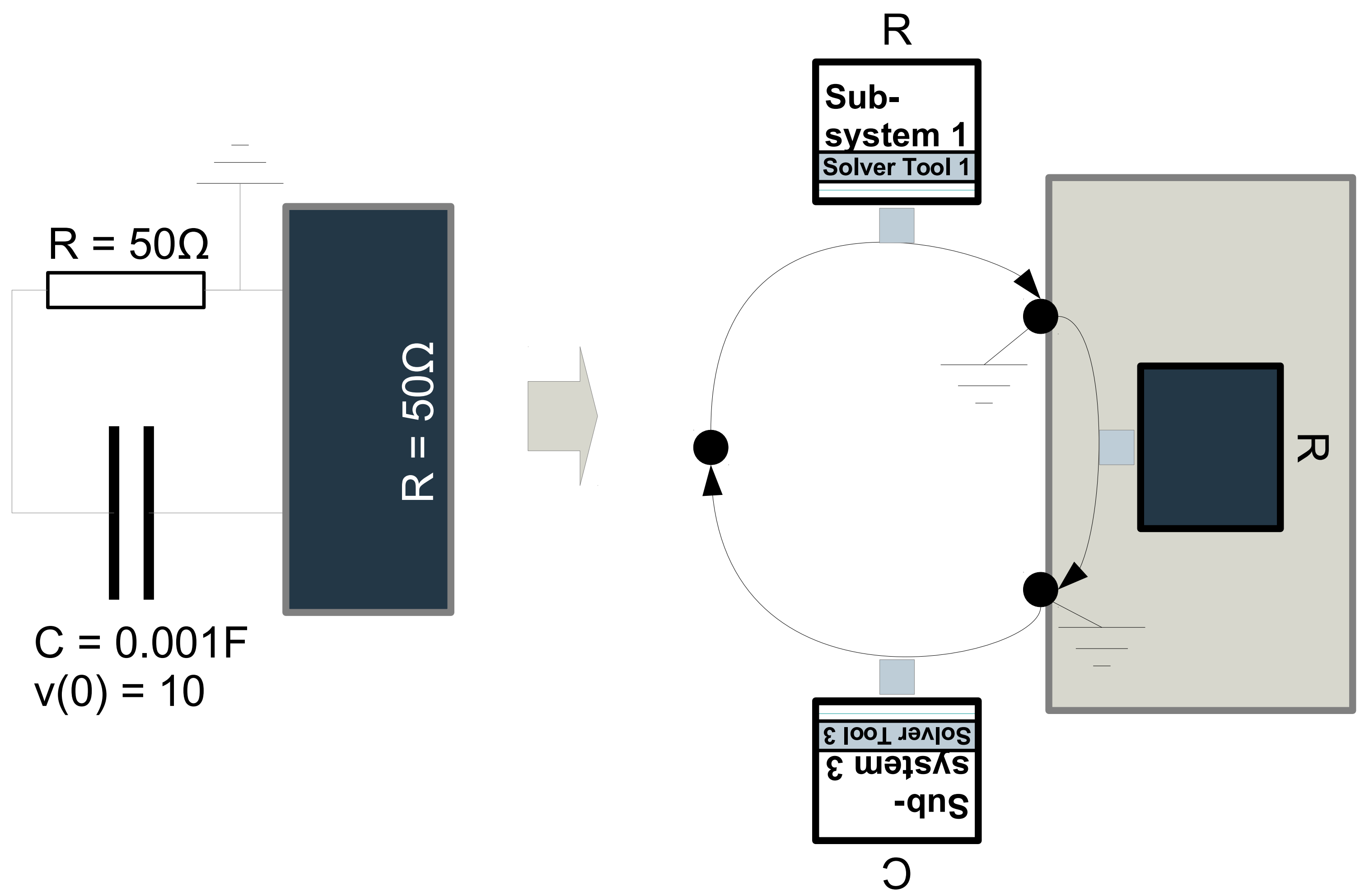


Notice also that **we can have this information at assembly time** without performing any equation processing!

Unrestricted © Siemens AG 2015 All rights reserved.

## Application of Linear Graphs to “FMI with physical ports”

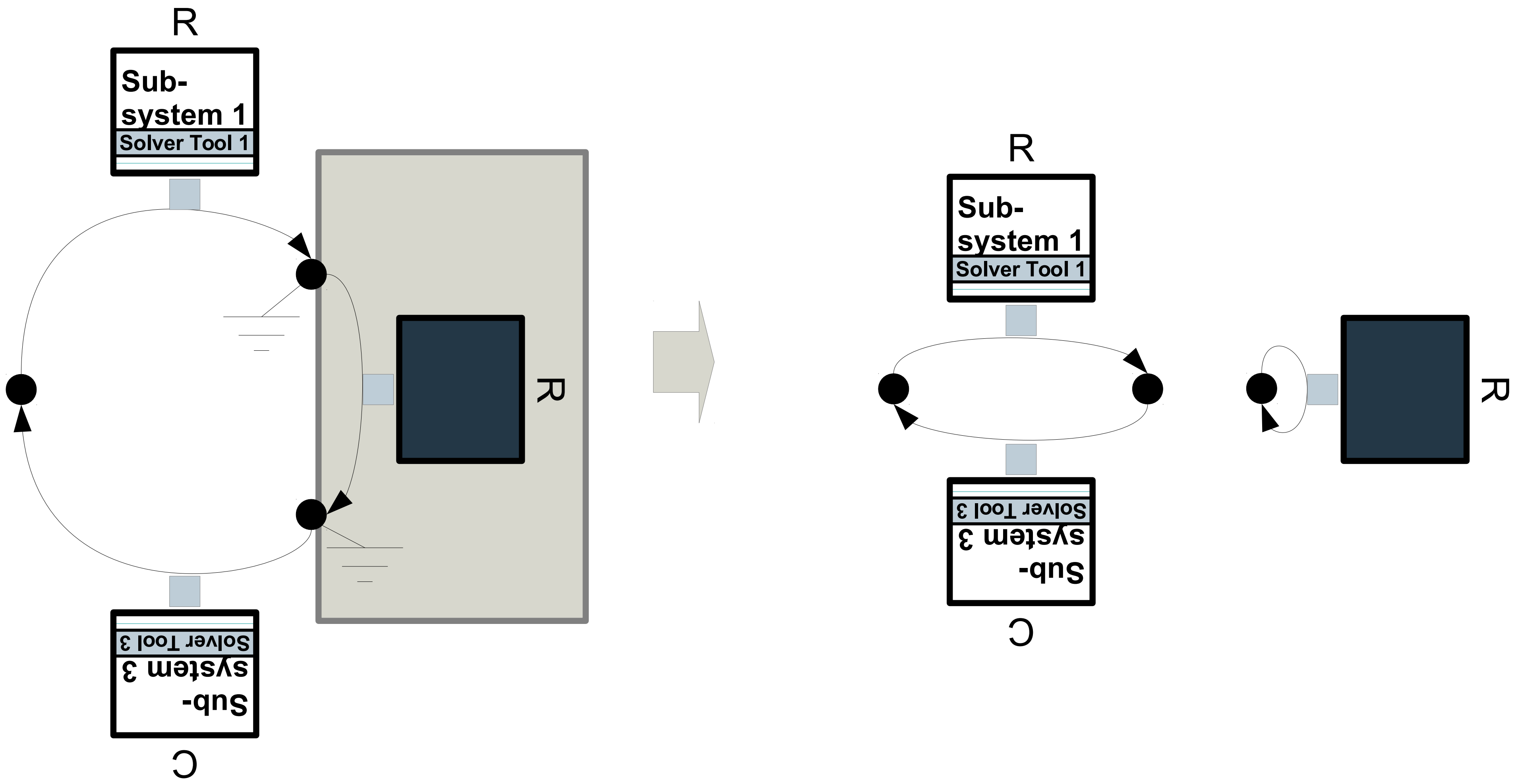
The second model can be defined like this:



Unrestricted © Siemens AG 2015 All rights reserved.

## Application of Linear Graphs to “FMI with physical ports”

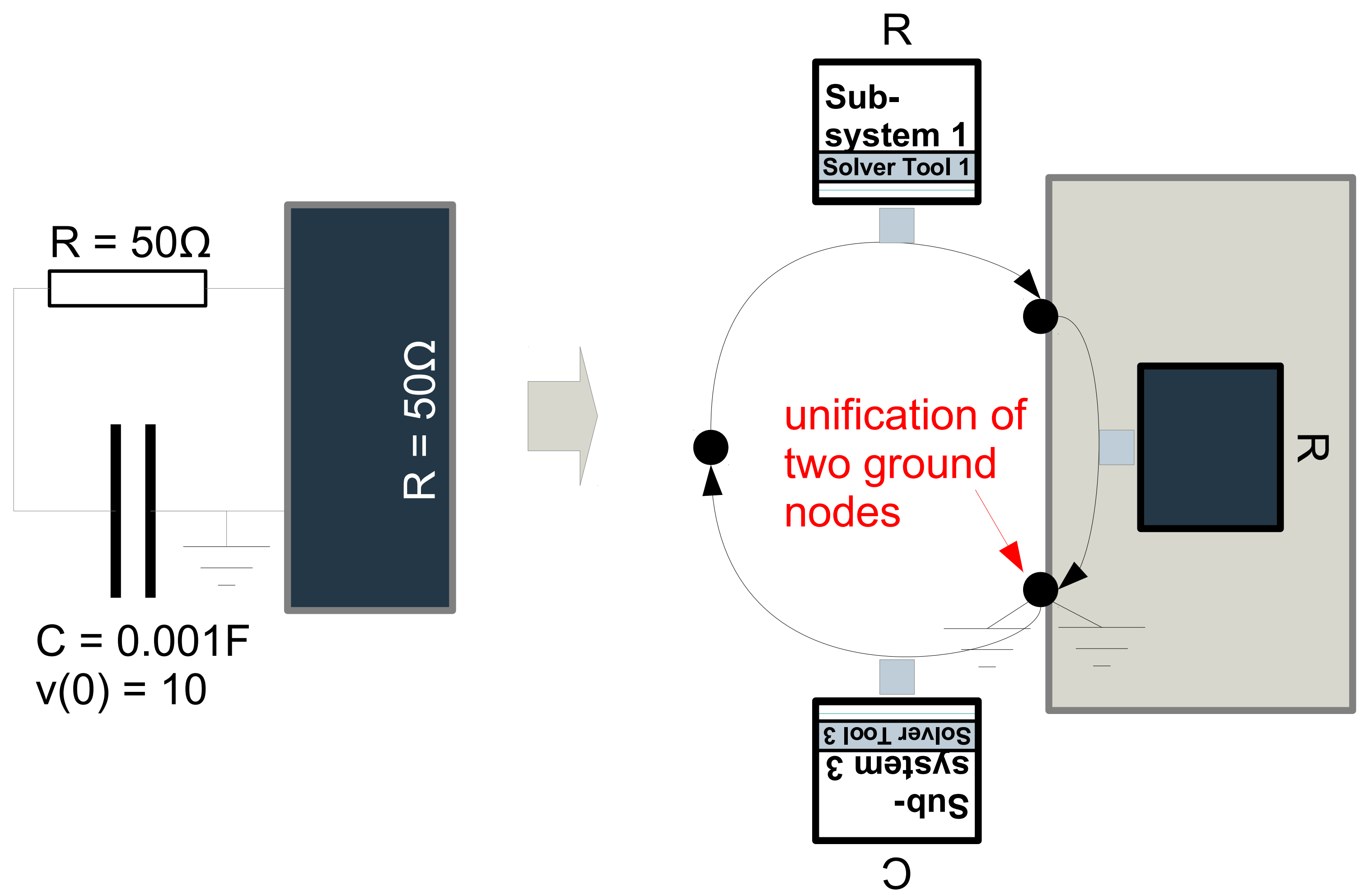
Again, we know as soon as assembly time that our model is not a serial circuit:



SIEMENS

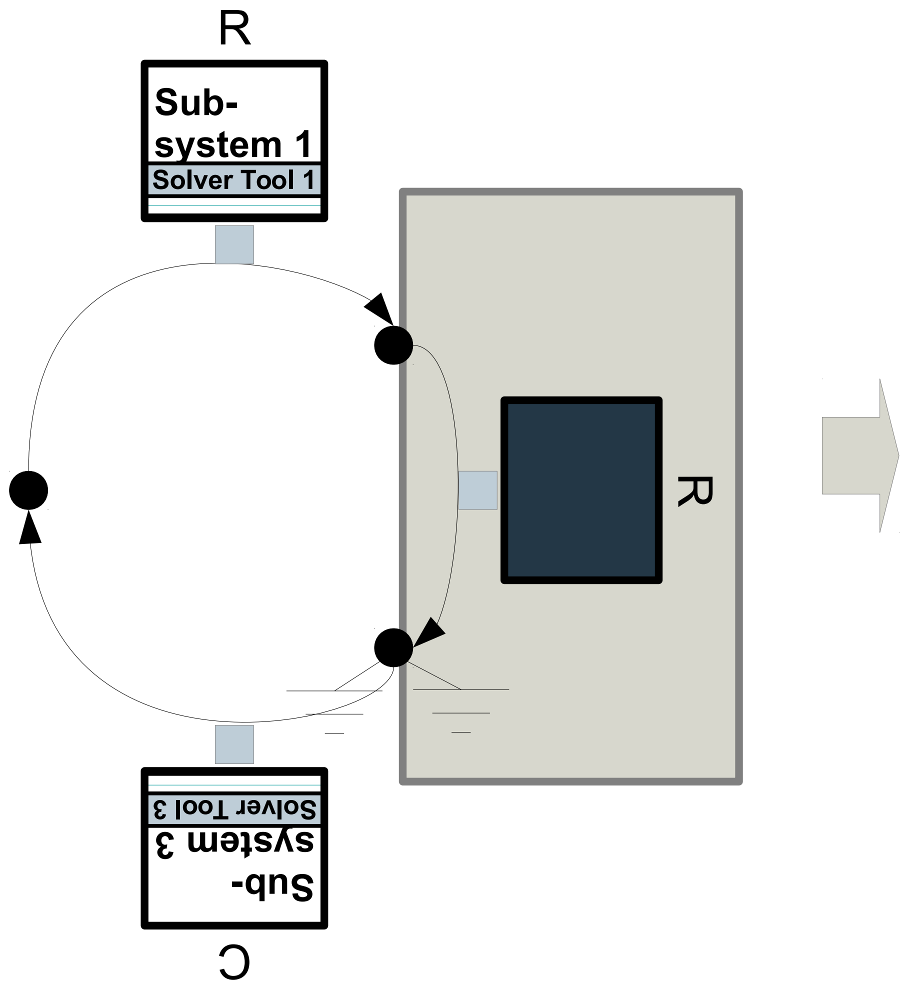
## Application of Linear Graphs to “FMI with physical ports”

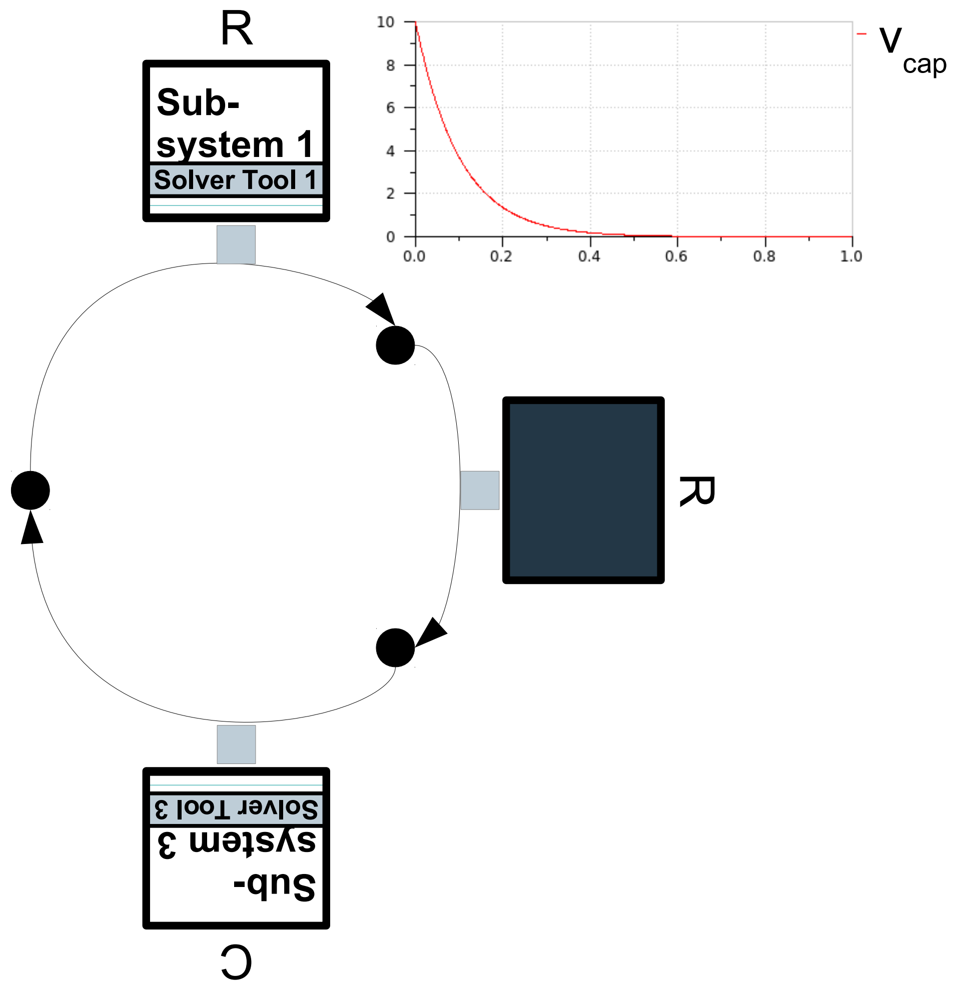
The third model can be defined like this:



## Application of Linear Graphs to “FMI with physical ports”

Amazingly, instead of leading to a singular system of equations as in Modelica, we get the expected, correct result:





Unrestricted © Siemens AG 2015 All rights reserved.

## Authors

- Sébastien Furic

  Siemens PLM Software

  14 boulevard de Valmy

  42300 Roanne, France

  `sebastien.furic@siemens.com`

- Antoine Viel

  Siemens PLM Software

  14 boulevard de Valmy

  42300 Roanne, France

  `antoine.viel@siemens.com`

**Unrestricted © Siemens AG 2015 All rights reserved.**

# Thank you