The Integration of System Dynamics with SysML and MBSE

Ivan Taylor
Policy Dynamics Inc.
ivan@policydynamics.ca

Structure Determines Behaviour

Outline

1. Introduction
2. SysML and MBSE => Structure
3. System Dynamics (SD) => Behaviour
4. Integrating SD with SysML
5. Resilient System Design
6. Benefits of the Integrated Approach
7. Key Takeaways

Introduction

Complex Systems and Resilience

Importance of modelling system structure and behaviour over time

Resilience as adapting to and recovering from adversity

Purpose of the Presentation

Explore the integration of SD with SysML and MBSE

Emphasize designing resilient systems

SysML and MBSE ⇔ Structure Diagrams

SysML (Systems Modeling Language)

A standardized modelling language for systems engineering

Uses Diagrams: Block Definition, Internal Block, Activity, Sequence, etc.

MBSE (Model-Based Systems Engineering)

Shift from a document-centric to a model-centric approach

Enhances communication and reduces errors

System Dynamics (SD) ⬄ Behaviour Diagrams

HIGHLY INTERCONNECTED SYSTEM OF DIFFERENTIAL EQUATIONS SOLVED USING NUMERICAL METHODS

TELLS YOU WHERE YOU ARE TODAY, HOW YOU GOT HERE, AND WHERE YOU'RE HEADED IF NOTHING CHANGES

USES FEEDBACK LOOPS, STOCKS AND FLOWS, TIME DELAYS

Need for Integration of SysML and SD

Challenges in Modeling Complex Systems

- SysML and MBSE may capture structure but not behaviour

Advantages of Integration with SD

- Holistic view of both structure and behaviour

Integrating SD with SysML

Mapping SD Elements to SysML

- Stocks and Flows ⇔ Blocks and Activities
- Feedback ⇔Dependencies and Control Flows
- Differential Equations ⇔ Parametric Diagrams

Primary Languages for SD with SysML

- Modelica
- Simulink

Resilient System Design

Resilience - Ability of a system to avoid, withstand and recover from disruptions

Modeling Resilience

- SysML to identify vulnerabilities and design mitigation strategies
- SD to simulate adversities, adaptation and recovery processes

Benefits of the Integrated Approach

Combines structure and behaviour perspectives

Better prediction of system behaviour after adversity

Unified models facilitate stakeholder engagement

Challenges and Considerations

TOOL COMPATIBILITY AND INTEROPERABILITY

NEED FOR VALIDATION AND VERIFICATION

REQUIRES KNOWLEDGE OF MULTIPLE MODELLING METHODOLOGIES

Key Takeaways

IMPORTANCE OF INTEGRATING SD WITH SYSML

BENEFITS IN DESIGNING RESILIENT SYSTEMS

POTENTIAL FOR FURTHER RESEARCH AND DEVELOPMENT

ENCOURAGE INTEGRATED MODELLING APPROACHES

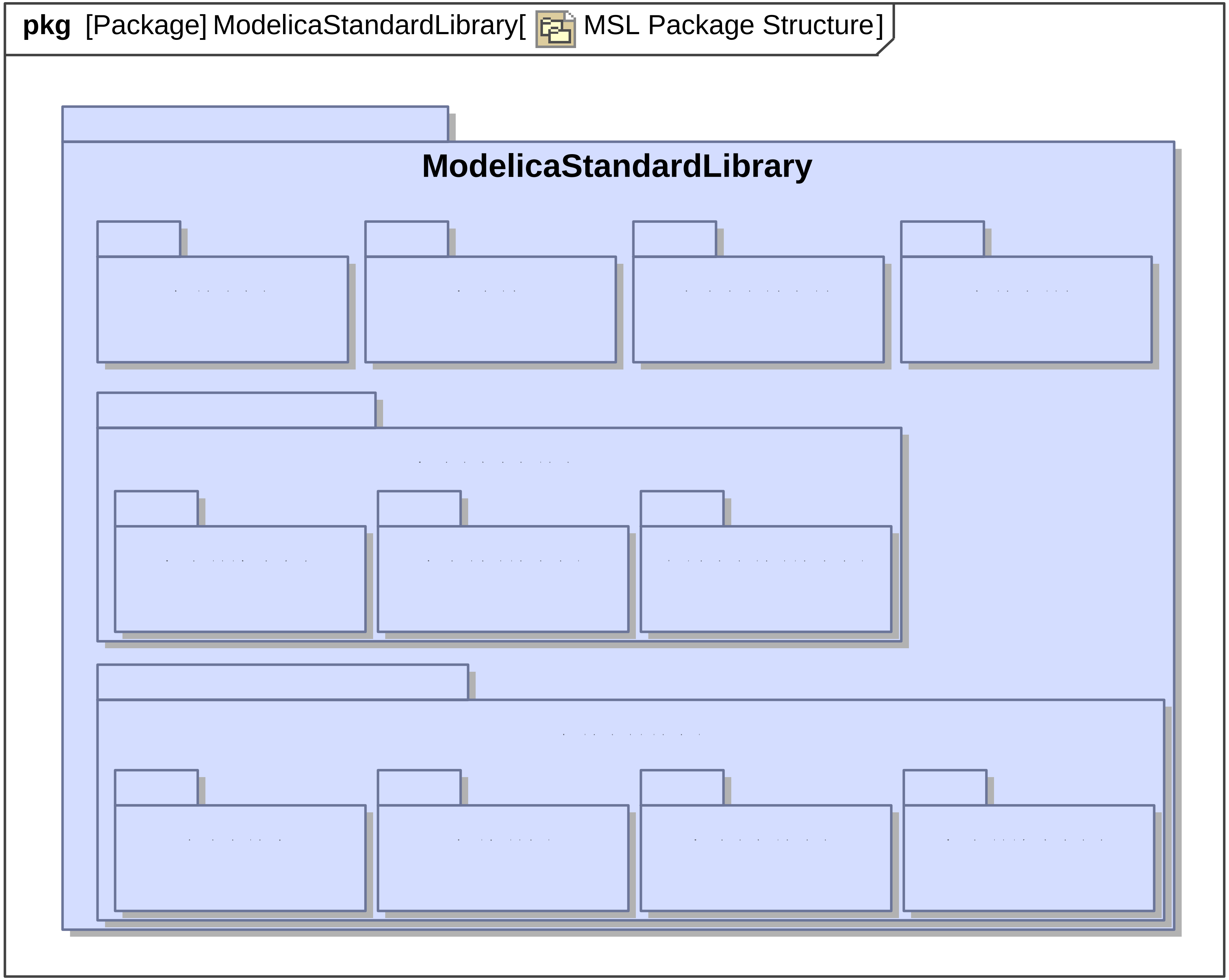
References

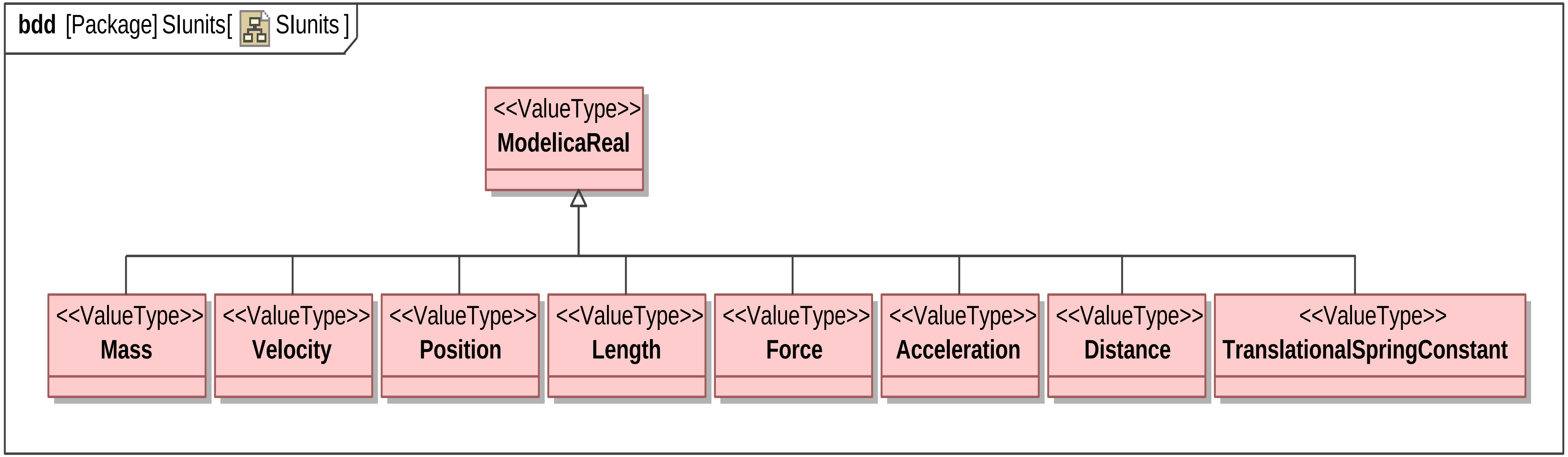
- Akbas, Asli Soyler, (2015), *Agent-Based and System Dynamics Hybrid Modeling and Simulation Approach Using Systems Modeling Language*, PhD Dissertation, University of Central Florida
- Johnson, Thomas, Paredis, , Christiaan J.J., and Burkhart, Roger, (2008) *Integrating Models and Simulations of Continuous Dynamics into SysML* The Modelica Association.
- Nikolaidou, M., Kapos, G., Tsadimas, A., Dalakas, V., & Anagnostopoulos, D. (2016). *Challenges in SysML Model Simulation*. *Advances In Computer Science: An International Journal, 5*(4), 49-56.
- Fu, Chao, Liu, Jihong, Yu, Hong Yan, Xu, WenTing, (2020), *A Visual transformation method of SysML model to Modelica model*, Journal of Physics: Conference Series 2020/11/01.
- Samares Engineering, (n/d), *Digital continuity between SysML and Simulink, Part 6, Advanced MBSE with SysML and other languages,* https://www.samares-engineering.com/en/2020/09/21/part-6-digital-continuity-between-sysml-and-simulink/
- Samares Engineering, (n/d), *Digital continuity between SysML and Modelica, Part 8, Advanced MBSE with SysML and other languages*, https://www.samares-engineering.com/en/2020/11/06/part-7-digital-continuity-between-sysml-and-modelica-copy/

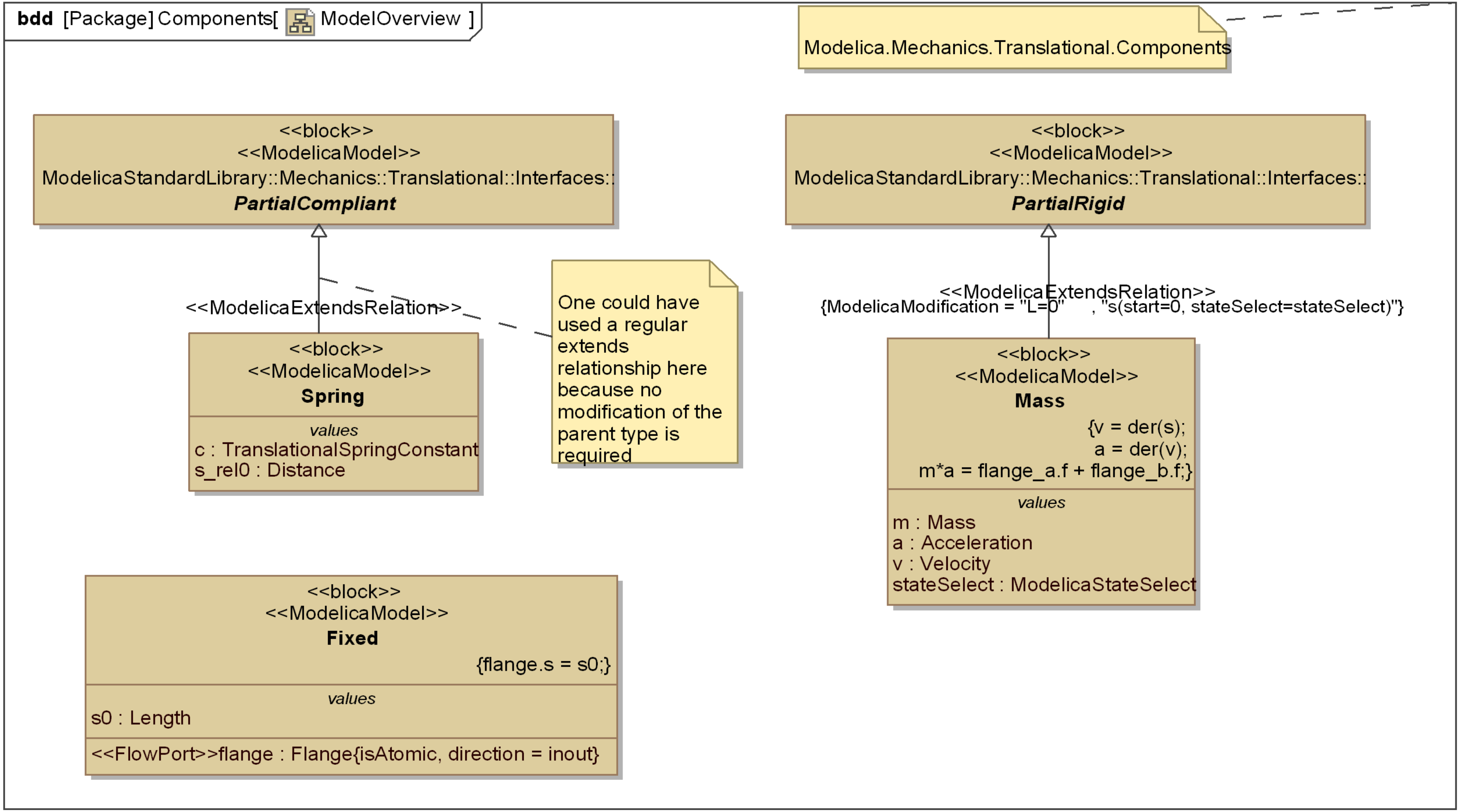
Thank you for your attention
Questions and Discussion

ivan@policydynamics.ca

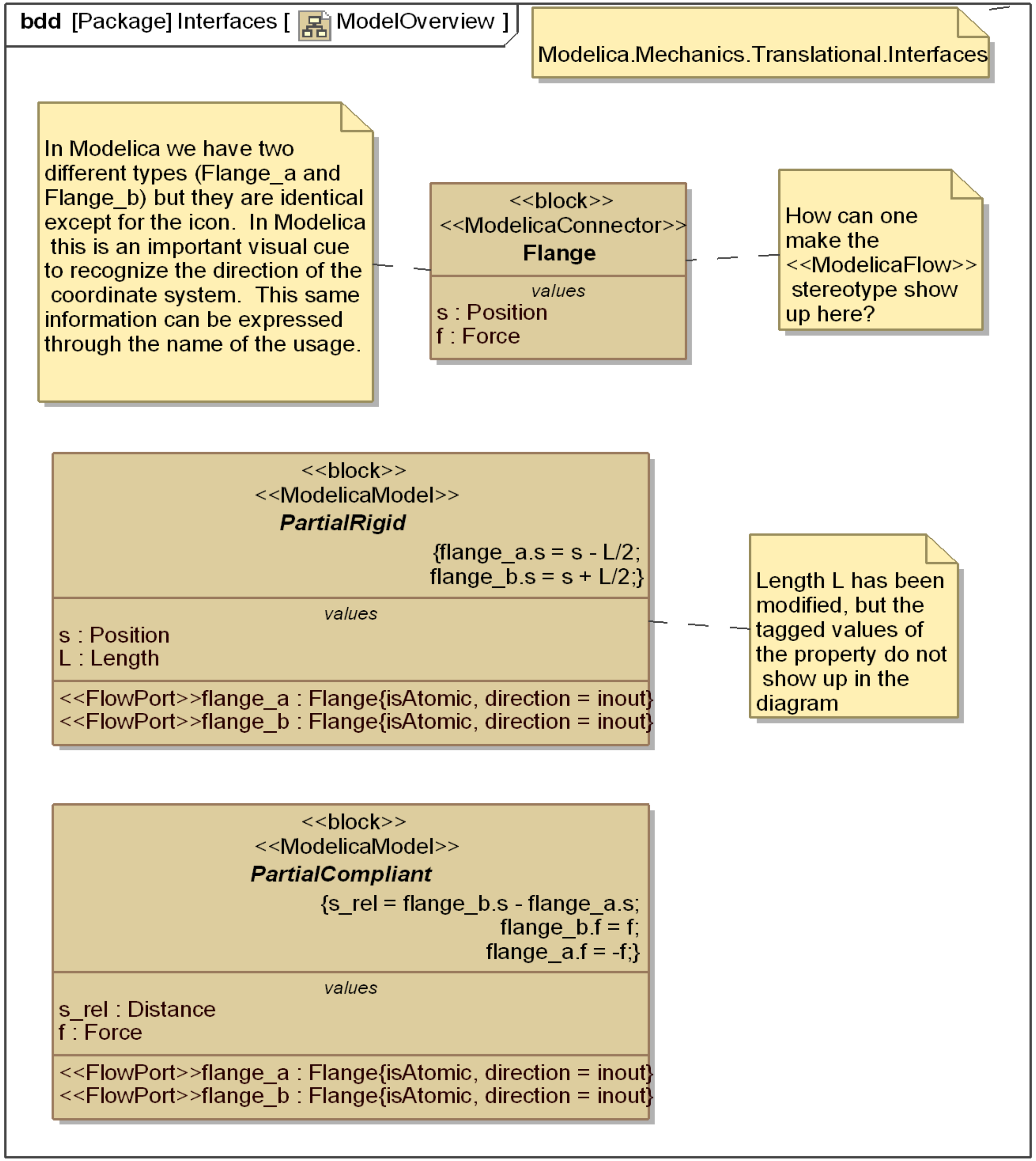
Oscillating Motion (Mass/Spring System)


pkg [Package] ModelicaStandardLibrary[MSL Package Structure]
ModelicaStandardLibrary
Blocks
Math
Constants
SIunits
Mechanics
MultiBody
Rotational
Translational
Electrical
Analog
Digital
Machines
MultiPhase


bdd [Package] SIunits[SIunits]
<<ValueType>>
ModelicaReal
<<ValueType>>
Mass
<<ValueType>>
Velocity
<<ValueType>>
Position
<<ValueType>>
Length
<<ValueType>>
Force
<<ValueType>>
Acceleration
<<ValueType>>
Distance
<<ValueType>>
TranslationalSpringConstant


bdd [Package] Components[ModelOverview]
Modelica.Mechanics.Translational.Components
<<block>>
<<ModelicaModel>>
ModelicaStandardLibrary::Mechanics::Translational::Interfaces::
PartialCompliant
<<block>>
<<ModelicaModel>>
ModelicaStandardLibrary::Mechanics::Translational::Interfaces::
PartialRigid
<<ModelicaExtendsRelation>>
<<ModelicaExtendsRelation>>
{ModelicaModification = "L=0" , "s(start=0, stateSelect=stateSelect)"}
<<block>>
<<ModelicaModel>>
Spring
values
c : TranslationalSpringConstant
s_rel0 : Distance
<<block>>
<<ModelicaModel>>
Mass
{v = der(s);
a = der(v);
m*a = flange_a.f + flange_b.f;}
values
m : Mass
a : Acceleration
v : Velocity
stateSelect : ModelicaStateSelect
<<block>>
<<ModelicaModel>>
Fixed
{flange.s = s0;}
values
s0 : Length
<<FlowPort>>flange : Flange{isAtomic, direction = inout}

One could have used a regular extends relationship here because no modification of the parent type is required



bdd [Package] Interfaces [ModelOverview]

Modelica.Mechanics.Translational.Interfaces

In Modelica we have two different types (Flange_a and Flange_b) but they are identical except for the icon. In Modelica this is an important visual cue to recognize the direction of the coordinate system. This same information can be expressed through the name of the usage.

<<block>>
<<ModelicaConnector>>
Flange

values
s : Position
f : Force

How can one make the <<ModelicaFlow>> stereotype show up here?

<<block>>
<<ModelicaModel>>
PartialRigid
{flange_a.s = s - L/2;
flange_b.s = s + L/2;}

values
s : Position
L : Length

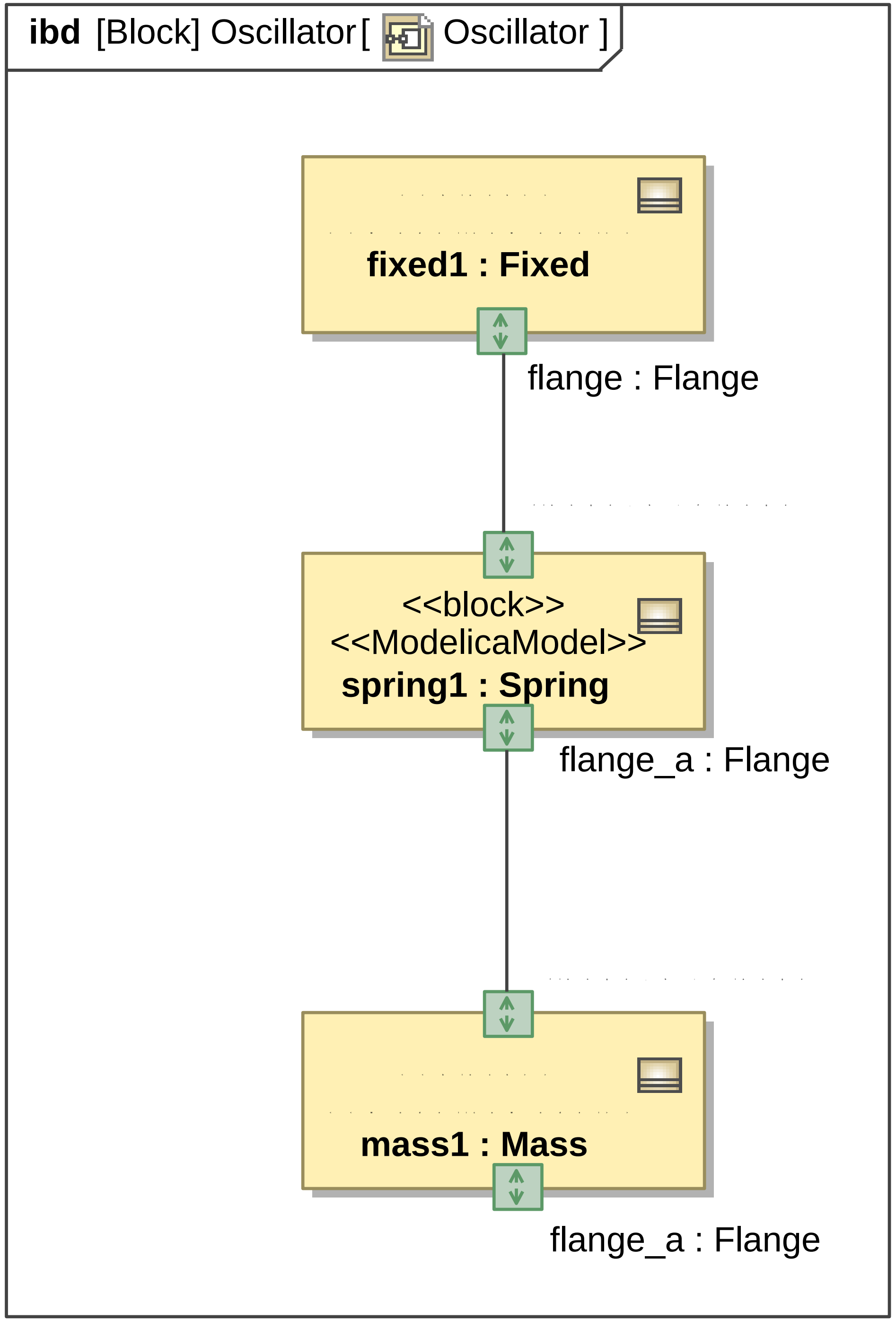
<<FlowPort>>flange_a : Flange{isAtomic, direction = inout}
<<FlowPort>>flange_b : Flange{isAtomic, direction = inout}

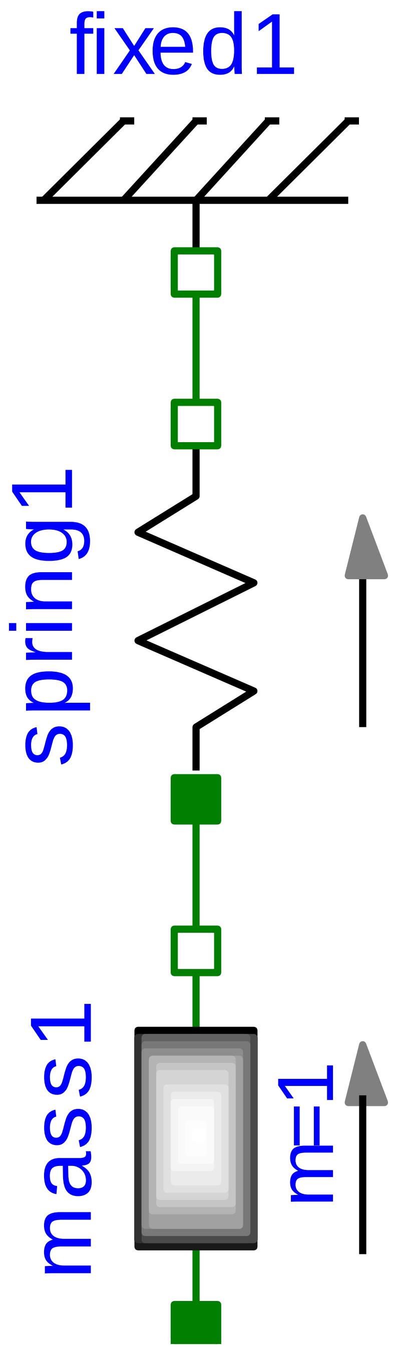
Length L has been modified, but the tagged values of the property do not show up in the diagram

<<block>>
<<ModelicaModel>>
PartialCompliant
{s_rel = flange_b.s - flange_a.s;
flange_b.f = f;
flange_a.f = -f;}

values
s_rel : Distance
f : Force

<<FlowPort>>flange_a : Flange{isAtomic, direction = inout}
<<FlowPort>>flange_b : Flange{isAtomic, direction = inout}


ibd [Block] Oscillator[Oscillator]
fixed1 : Fixed
flange : Flange
<<block>>
<<ModelicaModel>>
spring1 : Spring
flange_a : Flange
mass1 : Mass
flange_a : Flange


fixed1
spring1
mass1
m=1