

uri "<http://sabl.org/MinimalExampleMixed.sabl>" alias **mexmix**.

import "<http://sabl.org/ScientificConcepts1.sabl>".

Equation **velocityCalc**(**Position** **i**) returns **Velocity**: return **derivative**(**i**, ^**time**, 1).

Equation **accelerationCalc1**(**Velocity** **v**) returns **Acceleration**: return **derivative**(**v**, ^**time**, 1).

Equation **accelerationCalc2**(**Position** **i**) returns **Acceleration**: return **derivative**(**i**, ^**time**, 2).

Equation **momentumOfPhysicalObject**(**PhysicalObject** **o**) returns **Momentum**:
return **p**
where **p** is a **Momentum** with ^**value** (^**value** of **mass** of **o** * ^**value** of **velocity** of **o**),
with **unit** **unitResolver**("*", **unit** of **mass** of **o**, **unit** of **velocity** of **o**).

Rule **velocityOfPhysicalObject**:
if **o** is a **PhysicalObject**
then **velocity** of **o** is **velocityCalc**(**position** of **o**).

Rule **accelerationOfPhysicalObject1**:
if **o** is a **PhysicalObject**
then **acceleration** of **o** is **accelerationCalc1**(**velocity** of **o**).

Rule **accelerationOfPhysicalObject2**:
if **o** is a **PhysicalObject**
then **acceleration** of **o** is **accelerationCalc2**(**position** of **o**).

Rule **momentumOfPhysicalObjectRule**:
if **o** is a **PhysicalObject**
then **momentum** of **o** is **momentumOfPhysicalObject**(**o**).

Rule **newtons2ndLaw**:
if **o** is a **PhysicalObject** and **p** is **momentum** of **o** and
[**pv**, **pu**] is **derivative**(**p**, ^**time**, 1)
then there exists (a **Force** with ^**value** **pv**, with **unit** **pu**) and
force of **o** is the **Force**.