Modelica Change Proposal MCP-0024
Initialization of Clocked Discrete States
Status: Under Evaluation
2016-12-13, #2007, #1368

R. Franke

Summary

This MCP proposes to simplify initialization of clocked partitions and of clocked discretized continuous-time partitions by (a) introducing the new built-in operator firstTick() and by (b) allowing the built-in operators previous() and interval() in clocked discretized continuous-time partitions.

Revisions

Date	Description
Dec. 13, 2016	Completely revised by Martin Otter reflecting the discussion at the 90th Modelica Design Meeting (#2007, #1368)
June 20, 2016	Introduce generalized firstTick() operator
June 15, 2016	Update document structure Add HQP to test implementations, besides OpenModelica
June 13, 2016	Add section 1.2 Contradiction in Modelica spec 3.3 rev1, Update proposed change to Modelica spec
June 8, 2016	Initial version by Rüdiger Franke

Contributor License Agreement

Not yet applicable (since no CLA is available).

Table of Contents

1. Rationale

1.1 Simplifying the definition of start values at first clock tick

The previous values of discrete states in clocked partitions are initialized with their start values. See the following example:

```
model A
  Real x(start = 0.2);
equation
  when Clock(0.1) then
    x = previous(x) + 0.3;
  end when;
end A;
```

The previous value of the discrete state x, previous(x), takes the value 0.2 at the first clock tick. The discrete state x is computed to be 0.2+0.3 = 0.5 at the first clock tick.

Sometimes a modeler wants to initialize x and not previous(x). This is typically implemented with an additional Boolean state and a conditional expression for the first clock tick, like:

```
model B
  Real x(start = 0.1);
  Boolean first(start = true);
equation
  when Clock(0.1) then
    x = if previous (first) then 0.2 else previous(x) + 0.3;
    first = false;
  end when;
end B;
```

The discrete state x takes the value 0.2 at the first clock tick.

To simplify initialization and making it completely clear for the user, it is proposed to iintroduce a new built-in function firstTick() that returns true if the function is called at the first clock tick of the clocked partition where the function is called.

Example:

```
model B
  Real x(start = 0.1);
equation
  when Clock(0.1) then
    x = if firstTick() then 0.2 else previous(x) + 0.3;
  end when;
end B;
```

1.2 Simplifying clocked discretized continuous-time partitions

Some controllers have a continuous-time part that is discretized with a clock (e.g. inverse model of a plant) and a discrete part that is already described as discrete system, e.g. a discrete PI controller. Modelica 3.3 requirese to have these parts in two separate partitions which makes the code complicated. Example:

```
model C1
  Real u;
  Real y;
  Real z(start=0);
  Real x(start=2, fixed=true);
  Real xx;
  parameter Real T=0.1;
equation
  u = sample(sin(time));
  when Clock(Clock(0.1), solverMethod="ImplicitEuler") then
     der(x) = u;
  end when;
```

```
  xx = subSample(x,1);
  z = previous(z) + xx*(interval()/T);
  y = xx + z;
end TestPreviousC1;
```

Instead it is proposed that the built-in functions previous(..), interval(..), firstTick(..) are allowed to be used in clocked discretized continuous-time partition.

The above example simplifies then to:

```
model C2
  Real u;
  Real y;
  Real z(start=0);
  Real x(start=2, fixed=true);
  parameter Real T=0.1;
equation
  u = sample(sin(time));
  when Clock(Clock(0.1), solverMethod="ImplicitEuler") then
     der(x) = u;
     z = previous(z) + x*(interval()/T);
     y = x + z;
  end when;
end C2;
```

2. Proposed Changes in Specification

The precise text of the proposed changes with respect to Modelica Specification 3.3 are in the accompanying document MCP_0024_InitializationClockStates_SpecChanges.docx/.pdf.

3. Backward Compatibility

Since a new built-in operator is introduced, firstTick(..), the language is backward cmpatible, since such a function is not yet present. However, if a user already defined a function firstTick(..) this model will not be backwards compatible.

In Modelica 3.3 previous() and interval() are not allowed to be called in a clocked discretized continuous-time partition. Relaxing this restriction is therefore backwards compatible.

4. Test implementation

The firstTick() operator has been test implemented during the 90th Modelica Design Meeting in a local copy of OpenModelica. Dymola 2017 FD01 supports the firstTick(..) operator.

Both Dymola and OpenModelica did not implement the restriction that previous(..) and interval(..) cannot be called in clocked discretized continuous-time partitions. Therefore, this part of the proposal is already supported in both tools.

Futher tests have been conducted with the FMU importing tool **HQP**, enabling dynamic optimization of discrete-time models, no matter if the states are formulated as discrete states or as clocked continuous states.

5. References

The following ticket led to the introduction of a special rule for clocked states originating from continuous equations in revision 1 of Modelica spec 3.3:

https://trac.modelica.org/Modelica/ticket/1379

The following ticket discussed the issue for OpenModelica, including also the bad placement of the rule outside section 16.9 “Initialization of Clocked Partitions” and the contradiction between discrete and continuous clocked states:

https://trac.openmodelica.org/OpenModelica/ticket/3770