# What's new in Cameo Simulation Toolkit 18.2 FR
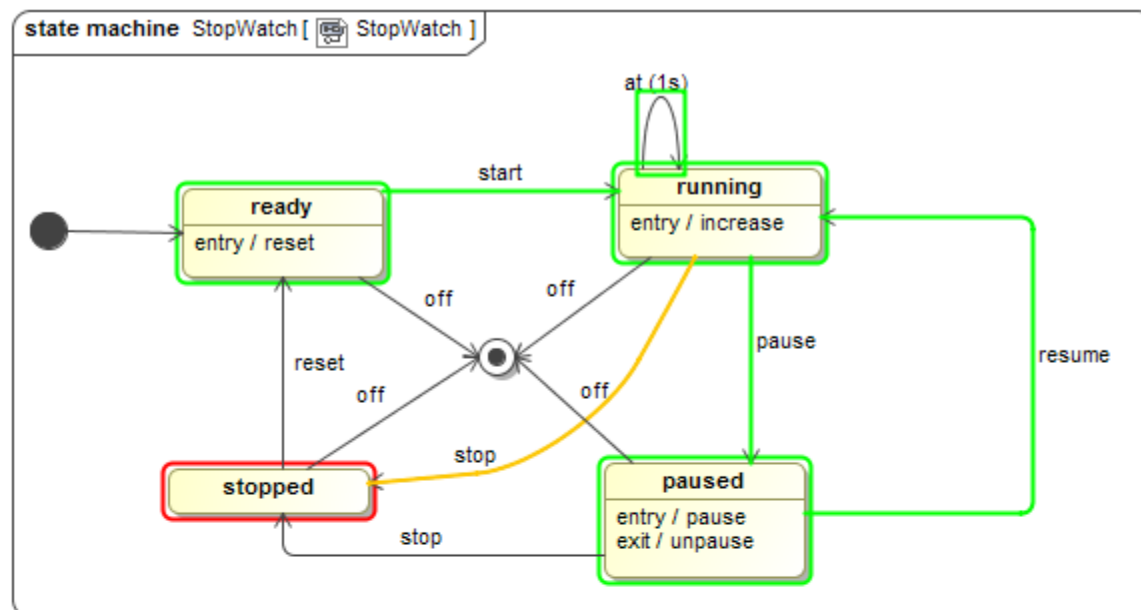
Timeline Plot

The new Timeline plot can be used to show how object states are changing over time in response to accepted events or stimuli. It implements UML Timing diagram concepts and additionally shows how long object was in particular state. The Timeline can be opened for any object in Variables panel or preconfigured in Simulation Configuration diagram.
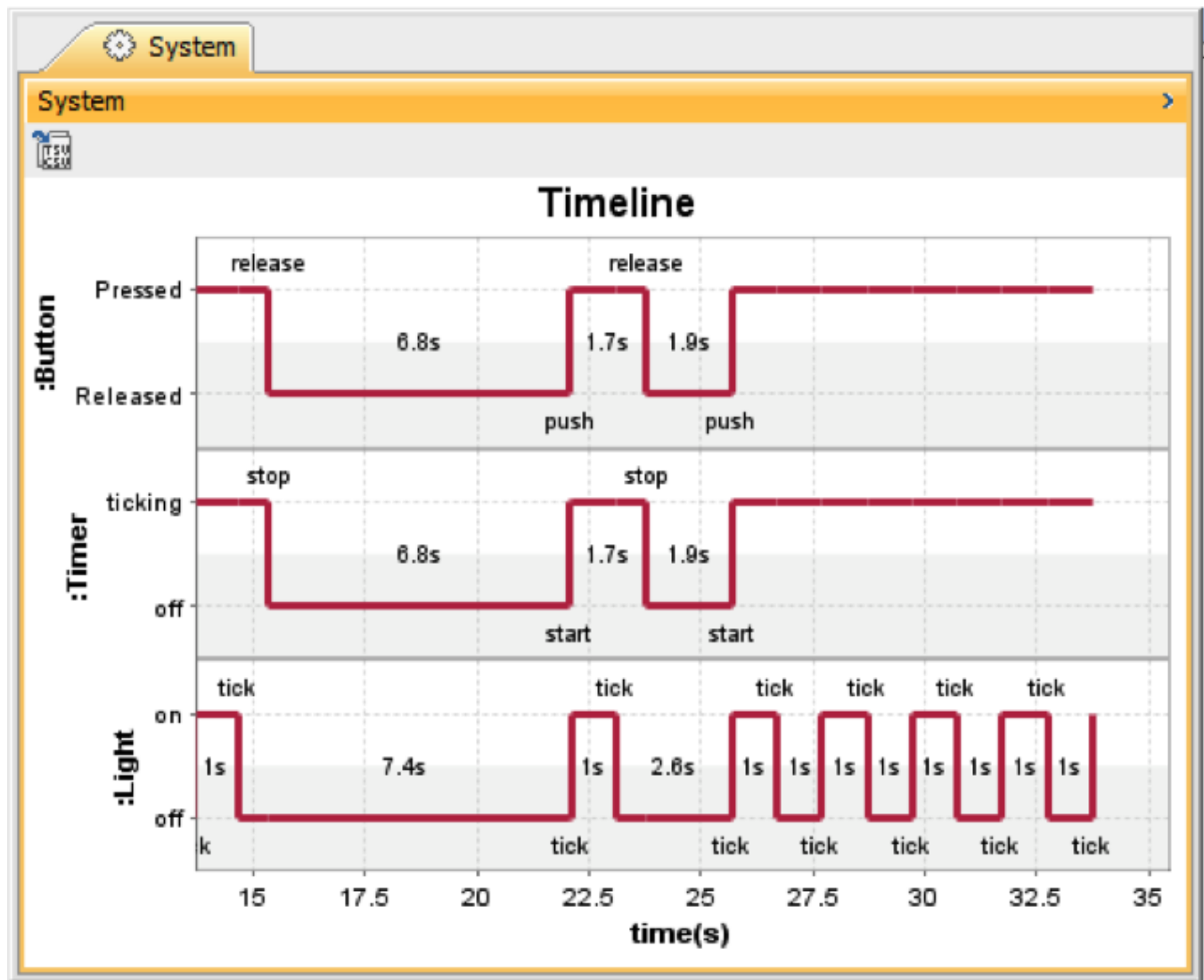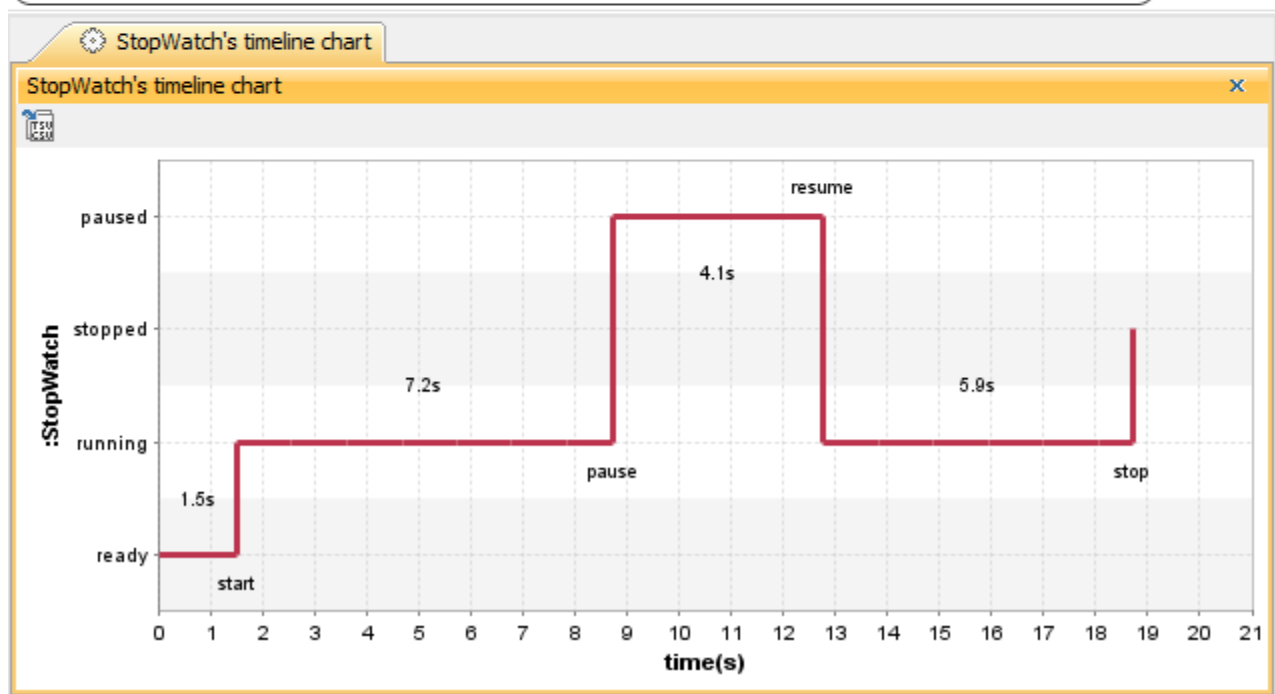
Log Filtering

To make Console output cleaner and more readable, the new logging options are introduced. The default **Warn** option leaves only start and finish messages and hides all unnecessary debug information.
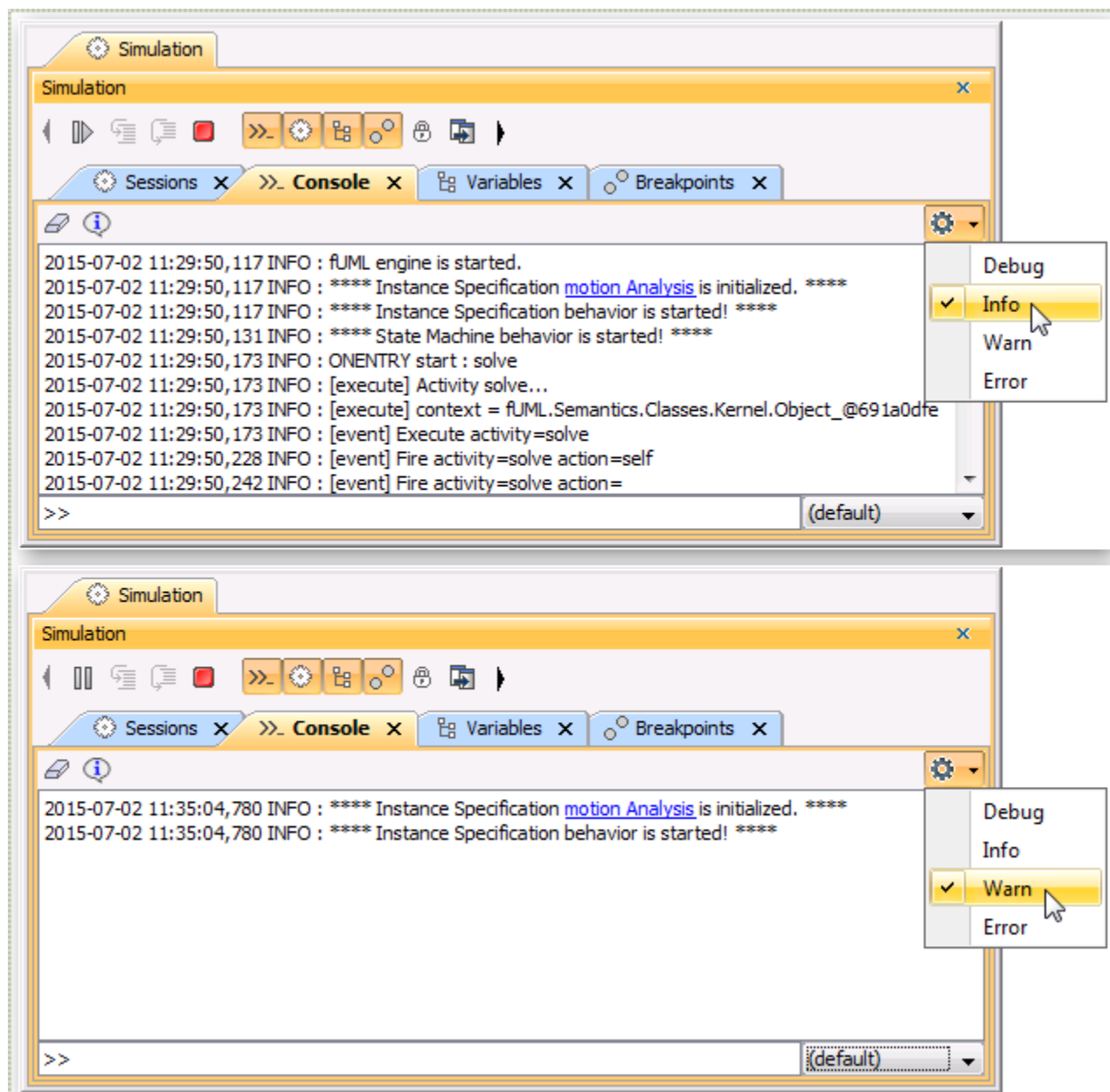
Evaluation of State Invariants

Sequence Diagrams become even more convenient testing tool by adding State Invariants support. If any of constraints on a lifeline is evaluated as *false*, execution of a testcase is immediately terminated with a *fail* verdict.

sd [Test Case] test_reset [test_reset]

```
          ┌─────────────────────┐              ┌─────────────────────┐
          │       «block»    📺 │              │  watch : StopWatch  │
          │ conductor : TestConductor │         │                     │
          └─────────────────────┘              └─────────────────────┘
                     ┆                                      ┆
                     │         1: reset                     │
          ┌──────────│──────────────────────────────────►  │
          │          █                                      █
          │          █                                      █
    {1 s} │          █                                      █
          │          █                                      █
          │          █     2: time1 = getTime()            █
          └──────────█──────────────────────────────────►  █
                     █                                      █
                     █                                      █
                     █                                      █
                     █
          ┌──────────────────────┐
          │   {time1 == 0}       │
          │                      │
          └──────────────────────┘
```
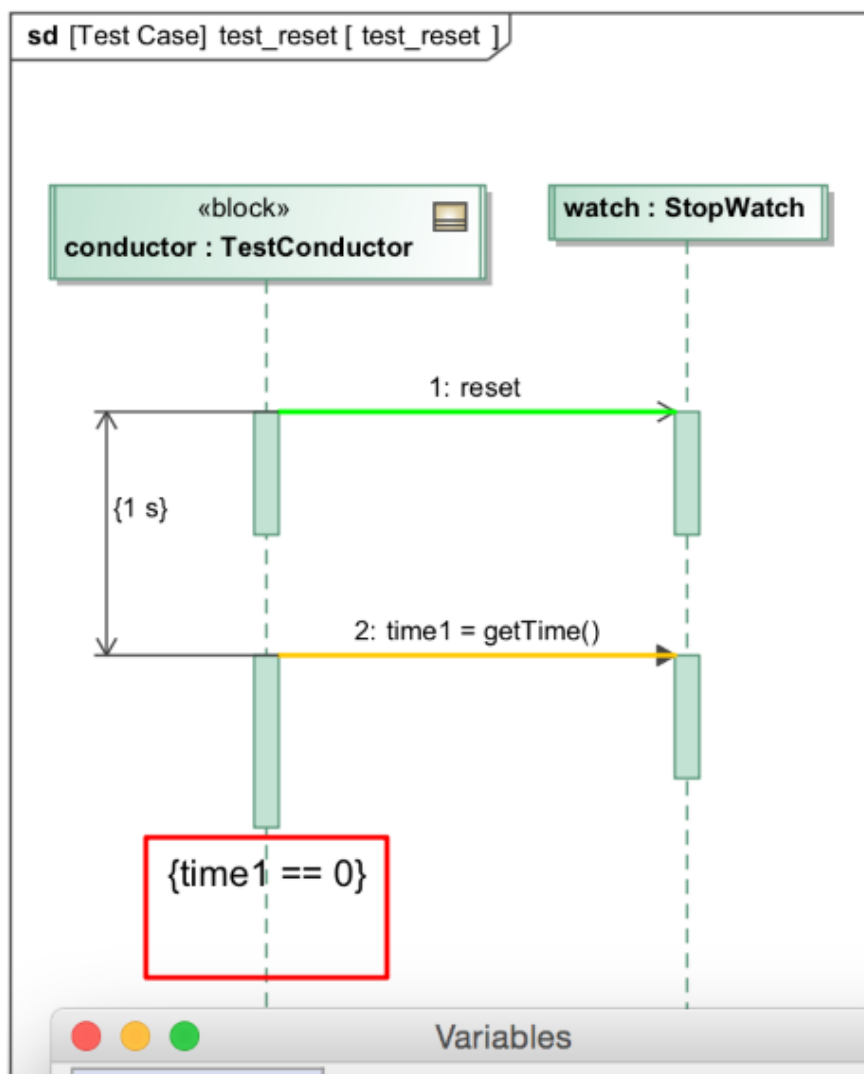
Variables

```
┌──────────────────────────────────────────────────────────────────┐
│ ● ● ●                      Variables                               │
├──────────────────────────────────────────────────────────────────┤
│ 🔲 Variables  ✕                                                   │
├──────────────────────────────────────────────────────────────────┤
│  🔄  🔽  🔽                                           ⚙ ▾         │
├────────────────────────────────────┬───────────────────────────── │
│  Name                              │ Value                         │
├────────────────────────────────────┼───────────────────────────── │
│  ⊟ 🔳 test_reset                   │ test_reset@64849faa           │
│       ● return verdict : VerdictKind│ fail                         │
├────────────────────────────────────┴───────────────────────────── │
│         Expand Recursively                                         │
│  ───────────────────────────────────                              │
│         Add Value                                                  │
│         Remove Value                                               │
│  ───────────────────────────────────                              │
│         Export Value To              ▶                            │
│  ───────────────────────────────────                              │
│         Go To                        ▶    ┌──────────────────────┬─────────────┐
│      🔳 Select in Containment Tree  ⌥B    │ Related Requirements ▶│ 🔳 Reset    │
│  ───────────────────────────────────      │ Related Testcases    ▶│             │
│         Edit                              └──────────────────────┴─────────────┘
└──────────────────────────────────────────┘
```
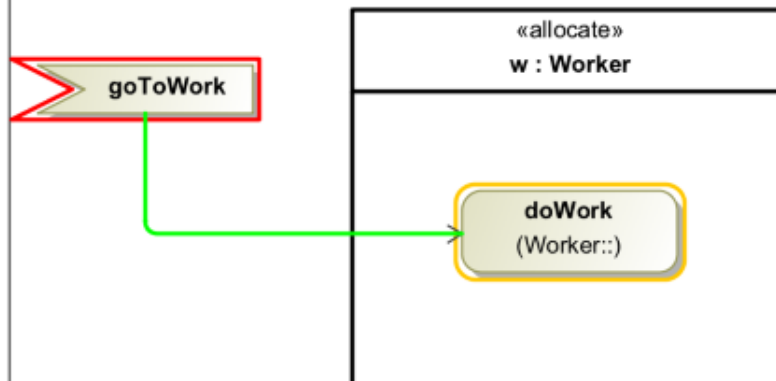
Call Event Support

The Call Event support is introduced in addition to already supported Signal Event, Time Event, and Change Event. Call Operation Action or Call Message which calls Operation without a method will generate a Call Event which can now be used on Triggers.
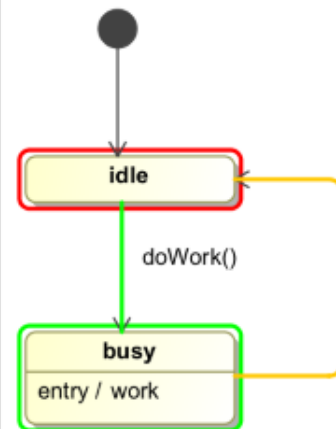
Swimlane Semantics

All actions inside a swimlanes are executed in a context of object represented by a swimlane partition. In case of multiple objects (multiplicity of represented part is more than 1) an action is allocated to the first non-busy object (which waits in particular state and does-nothing).
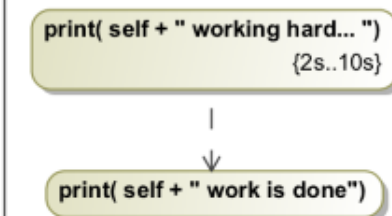
bdd [Package] system v1 [System]

act [Activity] system [🖳 system]

«allocate»
w : Worker

goToWork

doWork
(Worker::)

«block»
system

w   2..*

«block»
Worker

stm [State Machine] worker [🖳 worker]

idle

doWork()

busy

entry / work

act [Activity] work [🖳 work]

print(self + " working hard... ")
{2s..10s}

print(self + " work is done")

⚙ system

○ ○ ○   system

Workers

:Worker[1]

busy

6 s   0 s   4 s   0 s   9.9 s

idle

:Worker[2]

busy

9.2 s   0 s   8.5 s

idle

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40

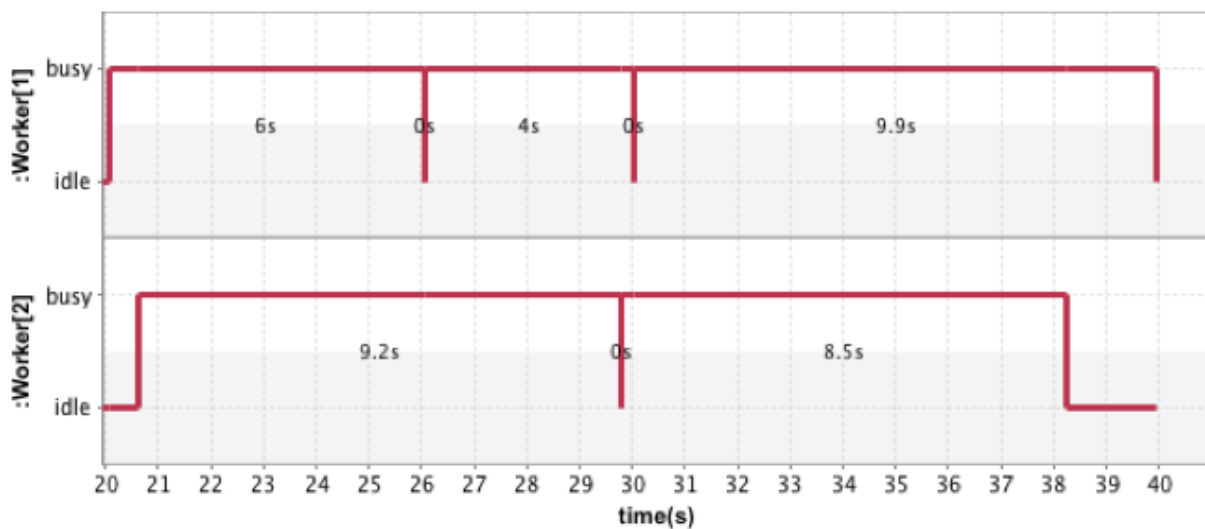**time(s)**

ValueTypes as Reals

All SysML Value types are now interpreted as Real values if it is not specified otherwise.