Exercises

1. This exercise studies the use of absent events in SR.

   (a) As a warmup, use Sequence and When to construct an SR model that generates a sequence of values *true* interspersed with *absent*. For example, produce the sequence

   $$(true, absent, absent, true, absent, true, true, true, absent)\ .$$

   Make sure your model adequately displays the output. In particular, *absent* should be visible in the display.

   (b) Create a composite actor **IsAbsent** that given any input sequence, produces an output *true* at every tick when the input is *absent*, and otherwise produces the output *absent*.

   (c) Create a composite actor that can recognize the difference between single and double mouse clicks. Your actor should have an input port named *click*, and two output ports, *singleClick* and *doubleClick*. When a `true` input at *click* is followed by *N* *absent*s, your actor should produce output `true` on *singleClick*, where *N* is a parameter of your actor. If instead a second `true` input occurs within *N* ticks of the first, then your actor should output a `true` on *doubleClick*.

   How does your model behave if given three values *true* within *N* ticks on input port *click*?

   (d) **Extra credit**: Redo (a)-(c) by writing a custom a Java actor for each of the three functions above. How does this design compare with the design implemented using primitive SR actors? Is it more or less understandable? complex?