Defining Visualizations' Interaction's Building Blocks

DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

Dario Benvenuti

With the help of:
Hao Cheng

Outline

1. Problem statement.
2. Why do we need this theory?
3. How could we build it?
4. Models of Atomic Widgets.
5. Example of use.
6. Questions.

Problem Statement

We want to build a clear, and rigorous, definition for **atomic widgets** by modeling them with the simplest FSA possible.

Having that, we could enable the possibility to combine those FSA together, to model more complex widgets, and, in the end, every existing visualization.

BUILDING BRICKS

Why?

1. Modeling every visualization from scratch can become costly and lead to ad-hoc models that work only for one particular visualization.

2. Well defined building bricks could lead to a great decrease in the complexity of interaction's modeling, leaving to the analyst/developer only to deal with the semantic.

3. With a really rigorous and event based definition, we could think of having, in the future, alghorithms for automated discovery of interaction's models (e.g., by parsing the source code and looking for visual objects with event handlers).

4. The FSA for some widget is so simple to be just a transition.

How?

Top-down approach, starting from visualizations:

We can model the interaction with a FSA.

It is a graph made up by states and transitions.

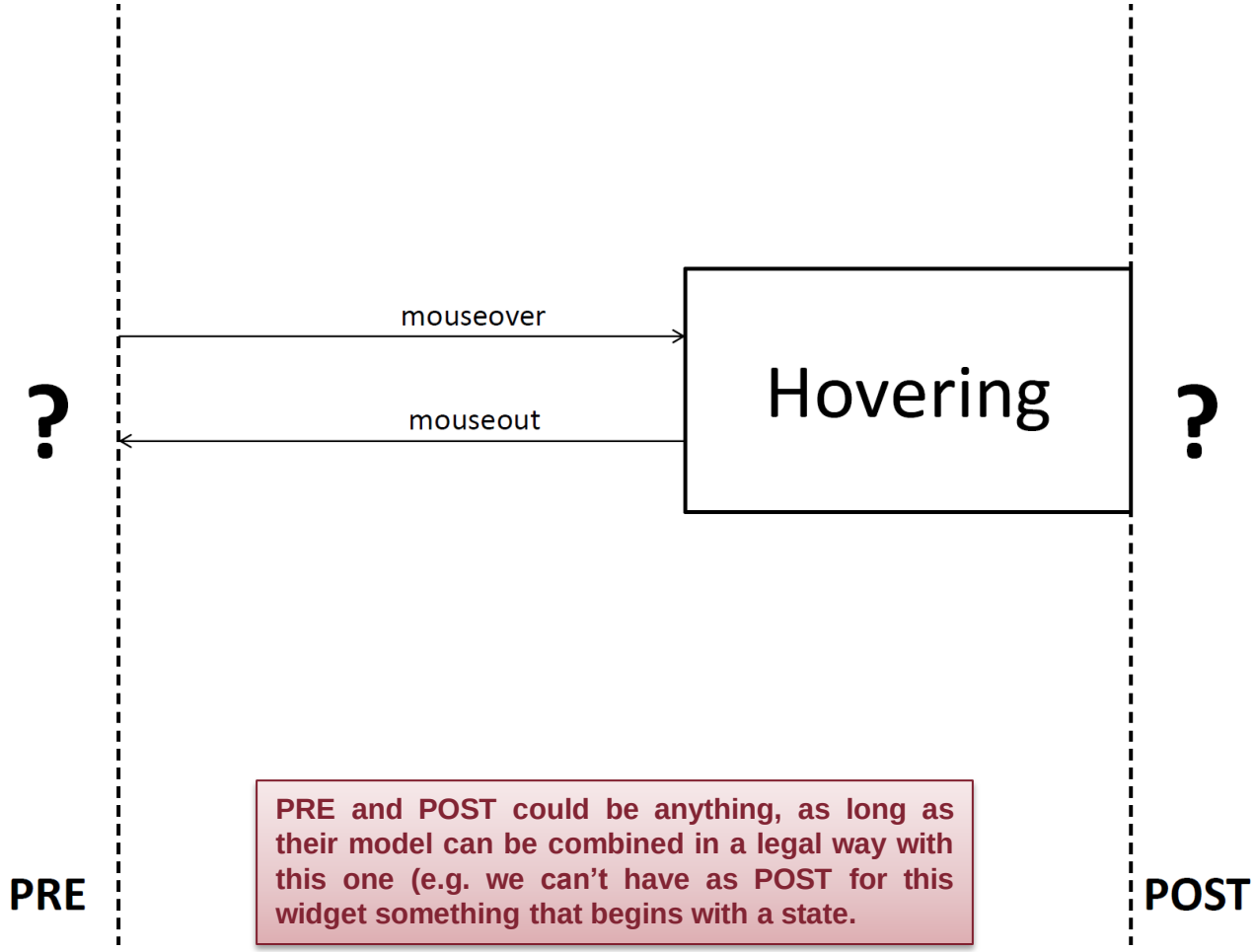
A widget could be a sub-graph of it, representing a particular sequence of steps in the interaction.

Then an **ATOMIC WIDGET** could be just a widget containing **AT MOST** one state and two transitions*, representing an atomic action that the user can perform.

* We need two transitions to enter and leave the state that can potentially be in the atomic widget. If there is no state, we could still have 2 transitions to represent widgets that can be «navigated» both ways.

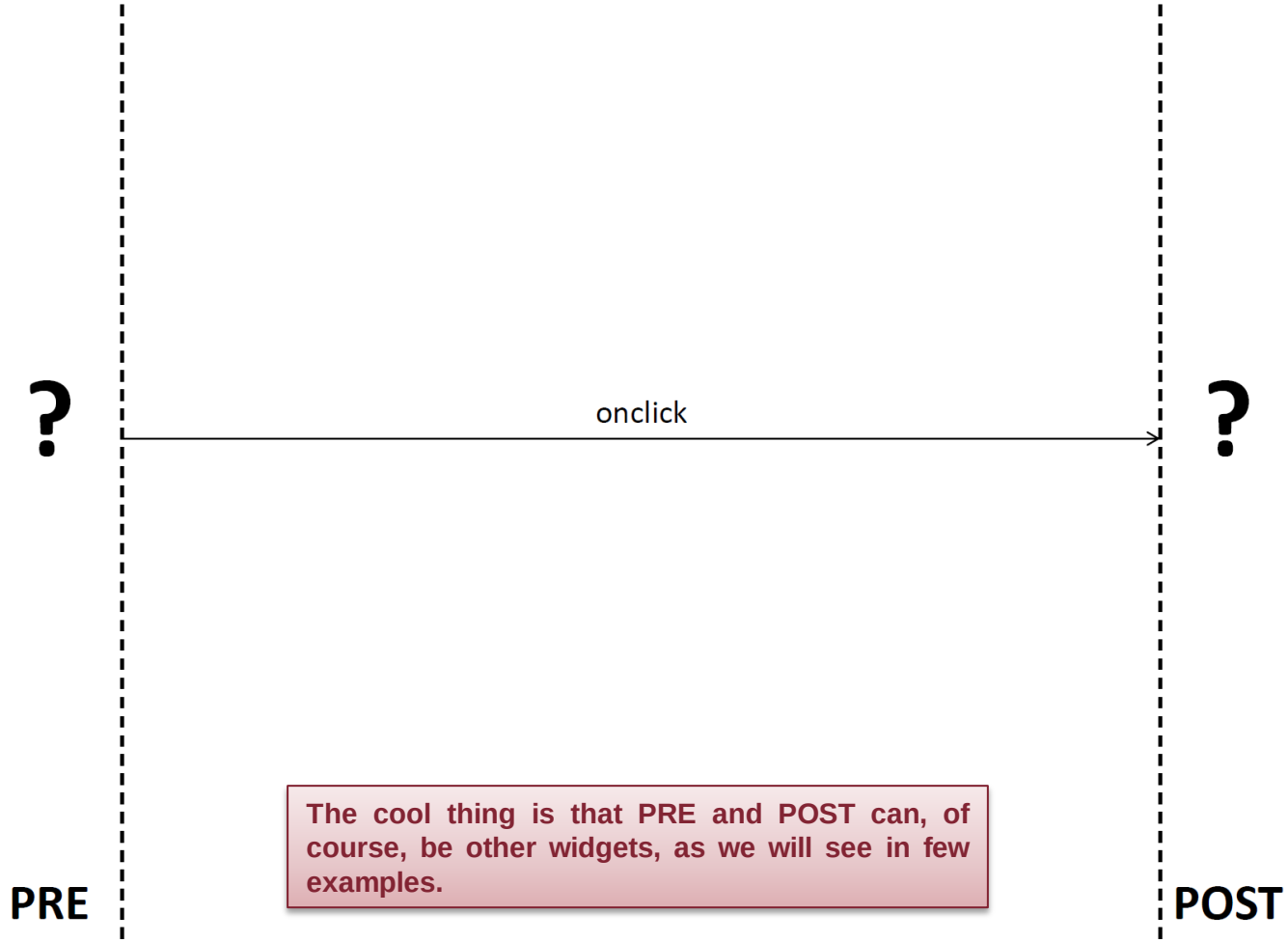
Atomic Widgets

Hovering



Atomic Widgets

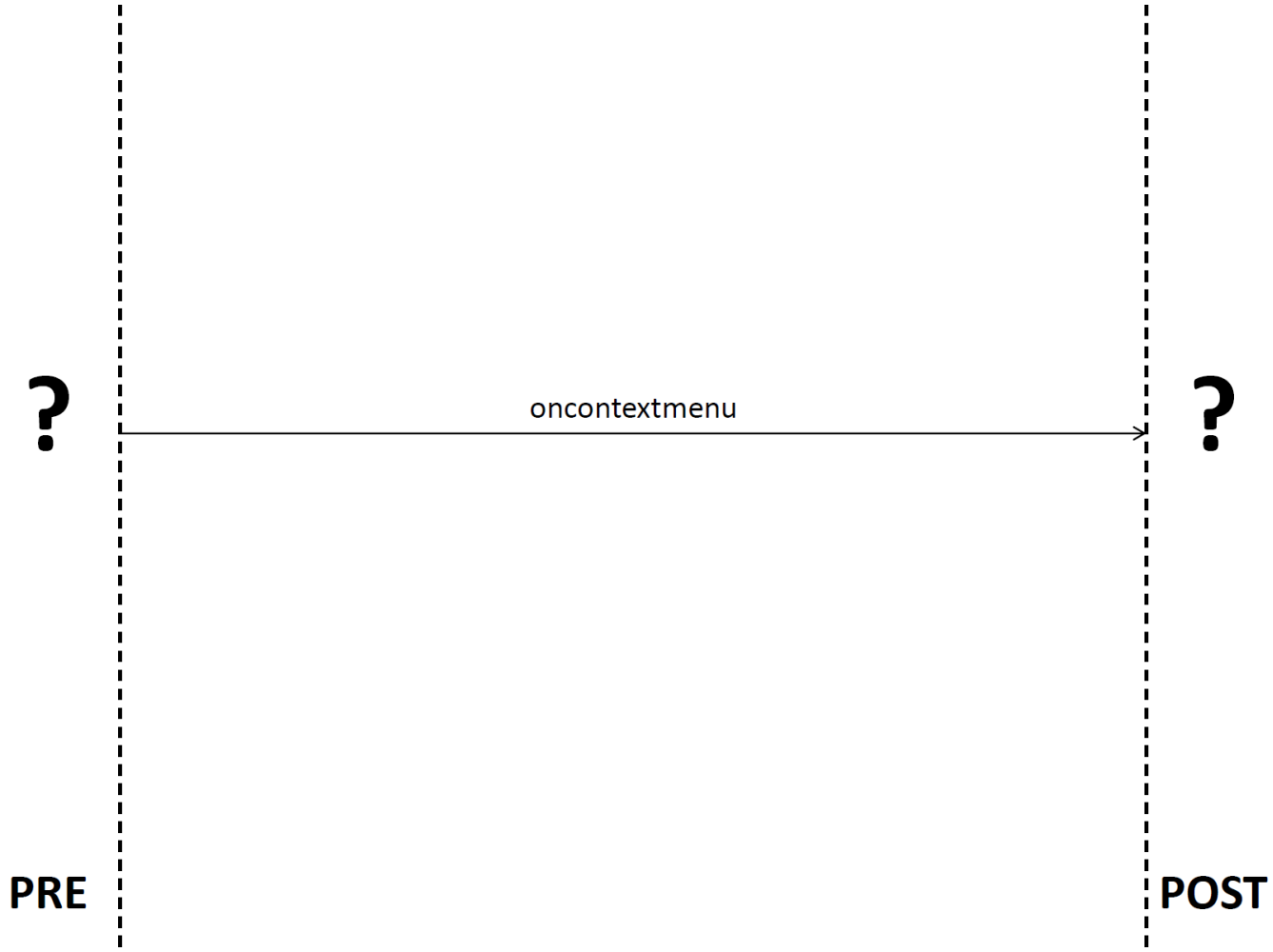
Click



The cool thing is that PRE and POST can, of course, be other widgets, as we will see in few examples.

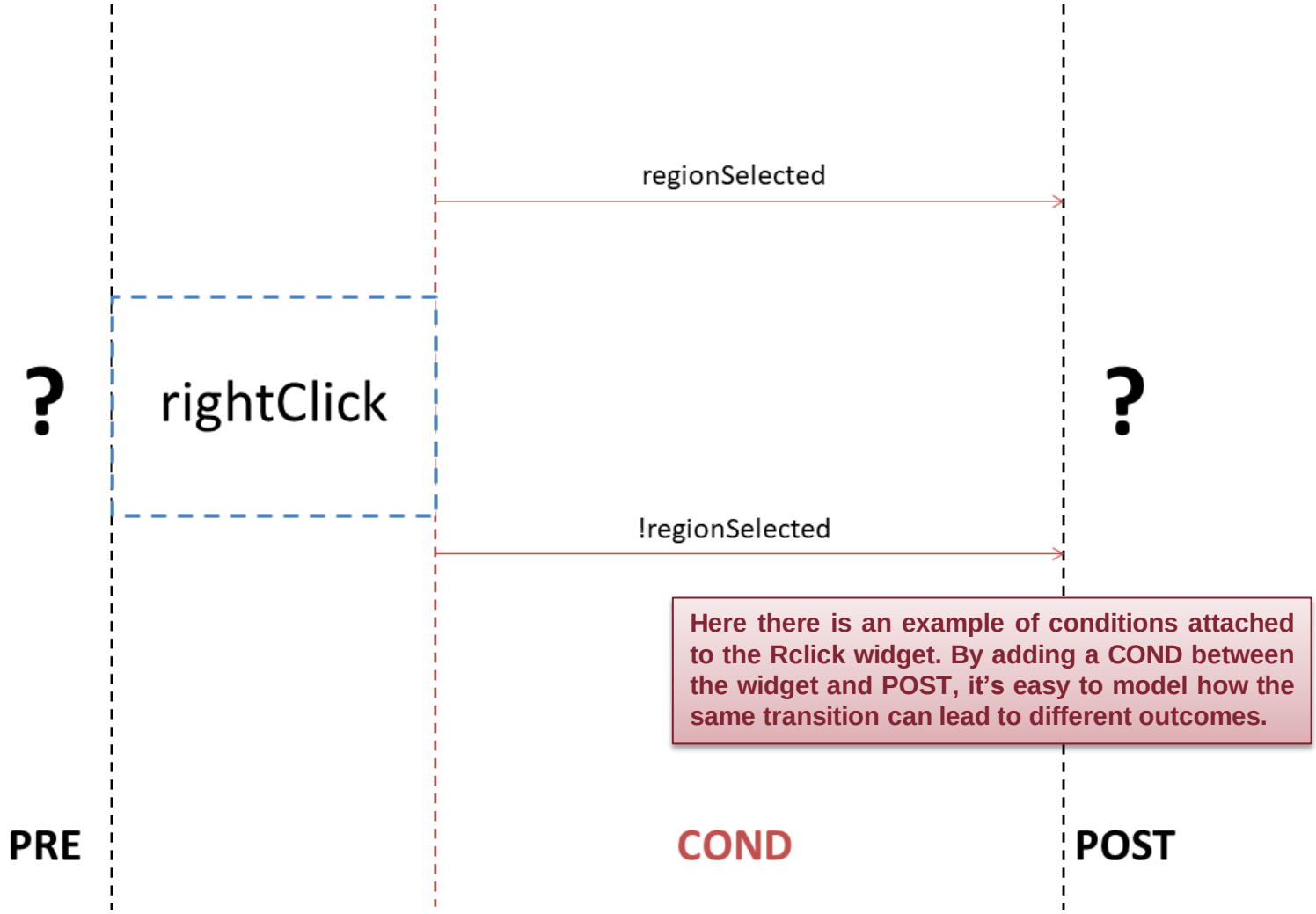
Atomic Widgets

RightClick



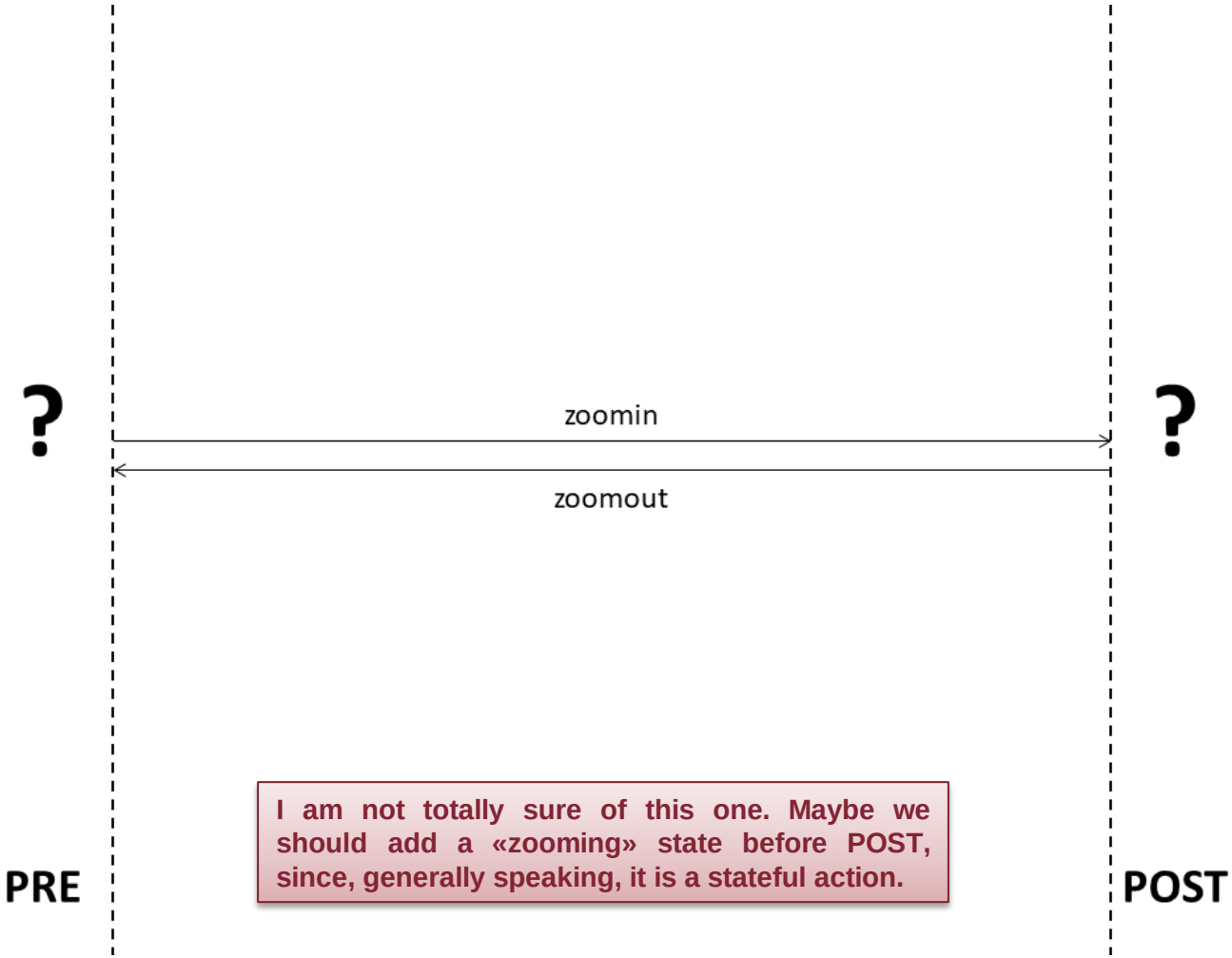
Atomic Widgets

RClick-Conditional



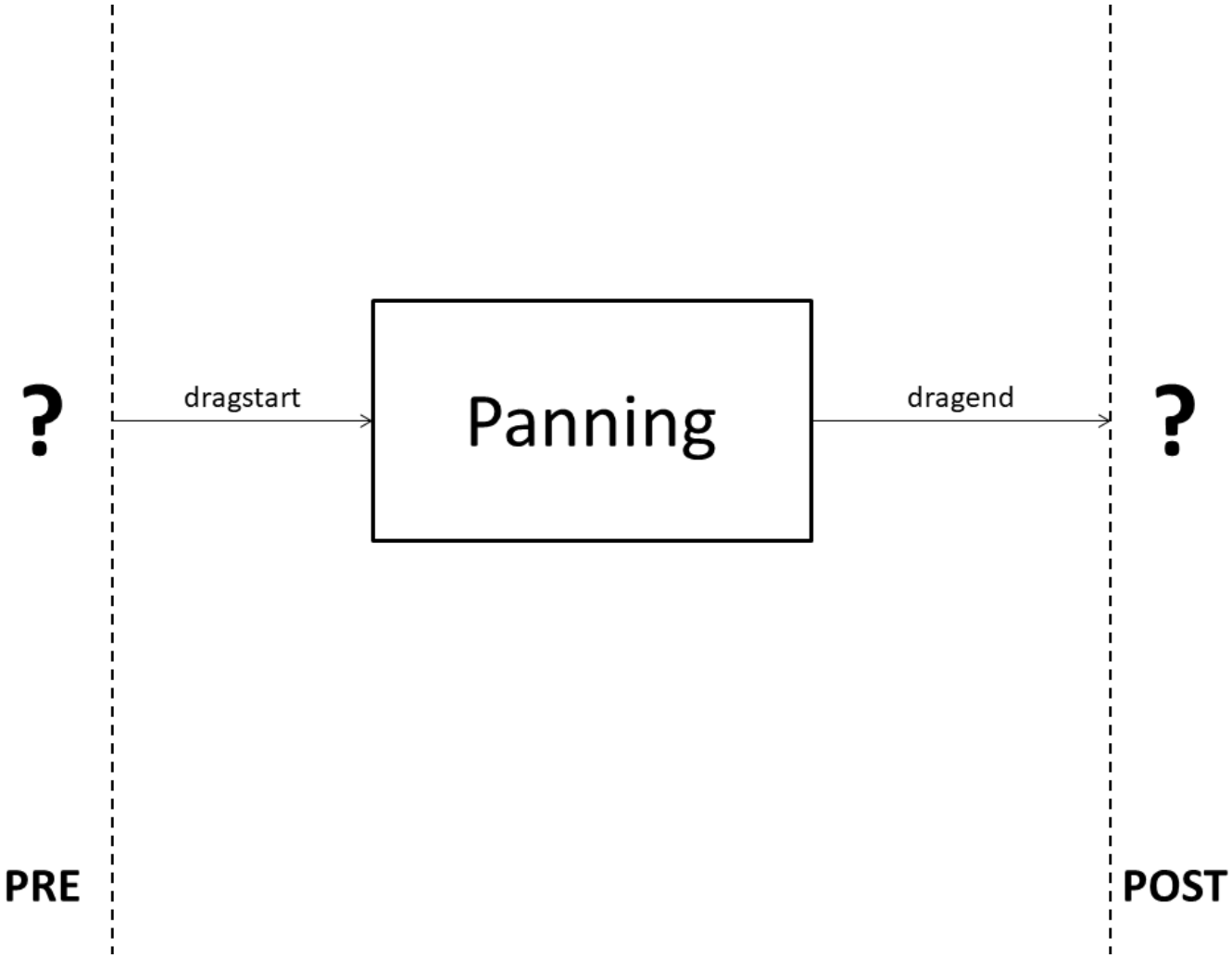
Atomic Widgets

Zoom

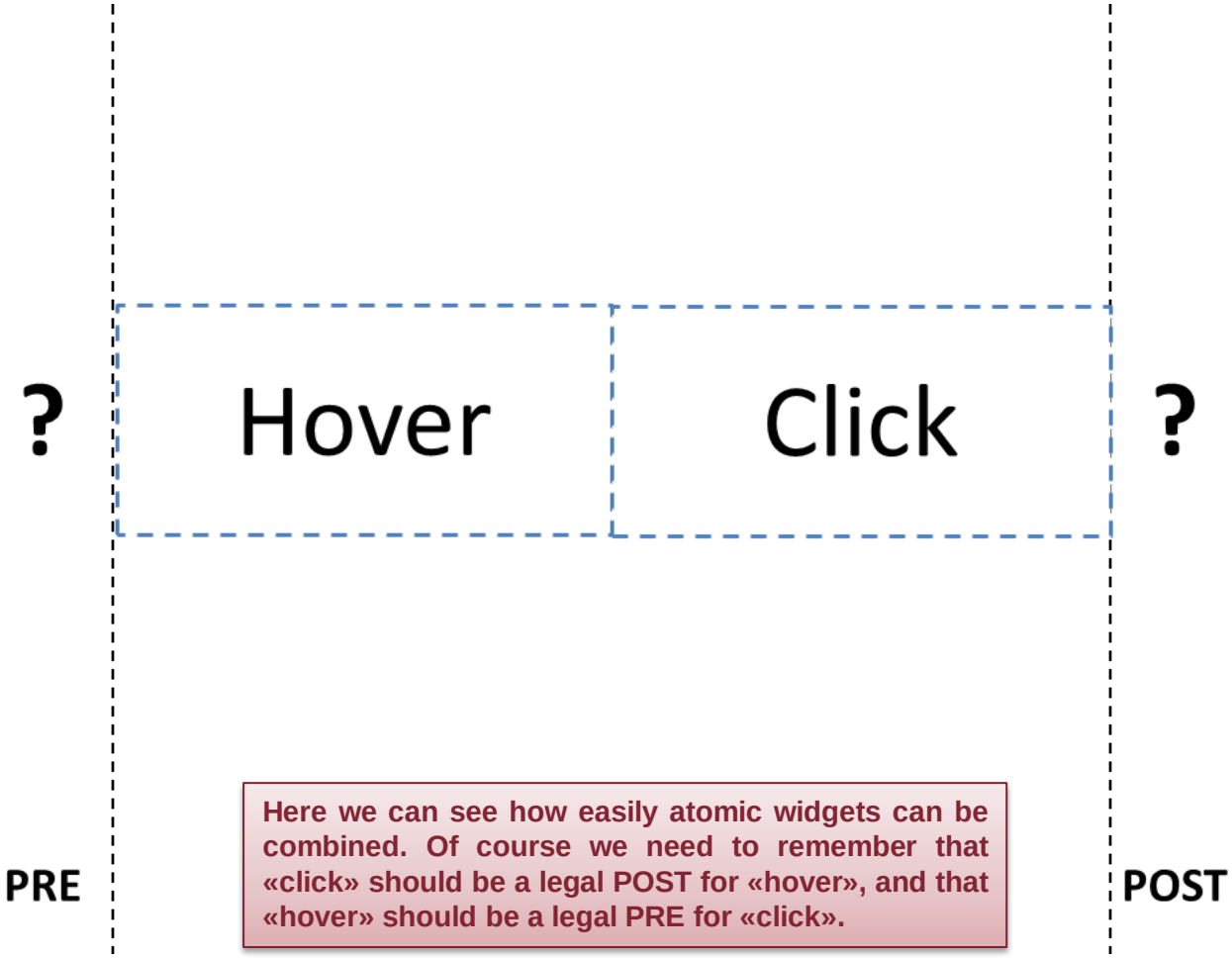


Atomic Widgets

Panning

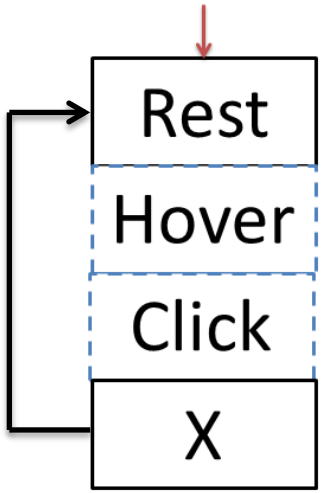


Example of Complex Widgets



Here we can see how easily atomic widgets can be combined. Of course we need to remember that «click» should be a legal POST for «hover», and that «hover» should be a legal PRE for «click».

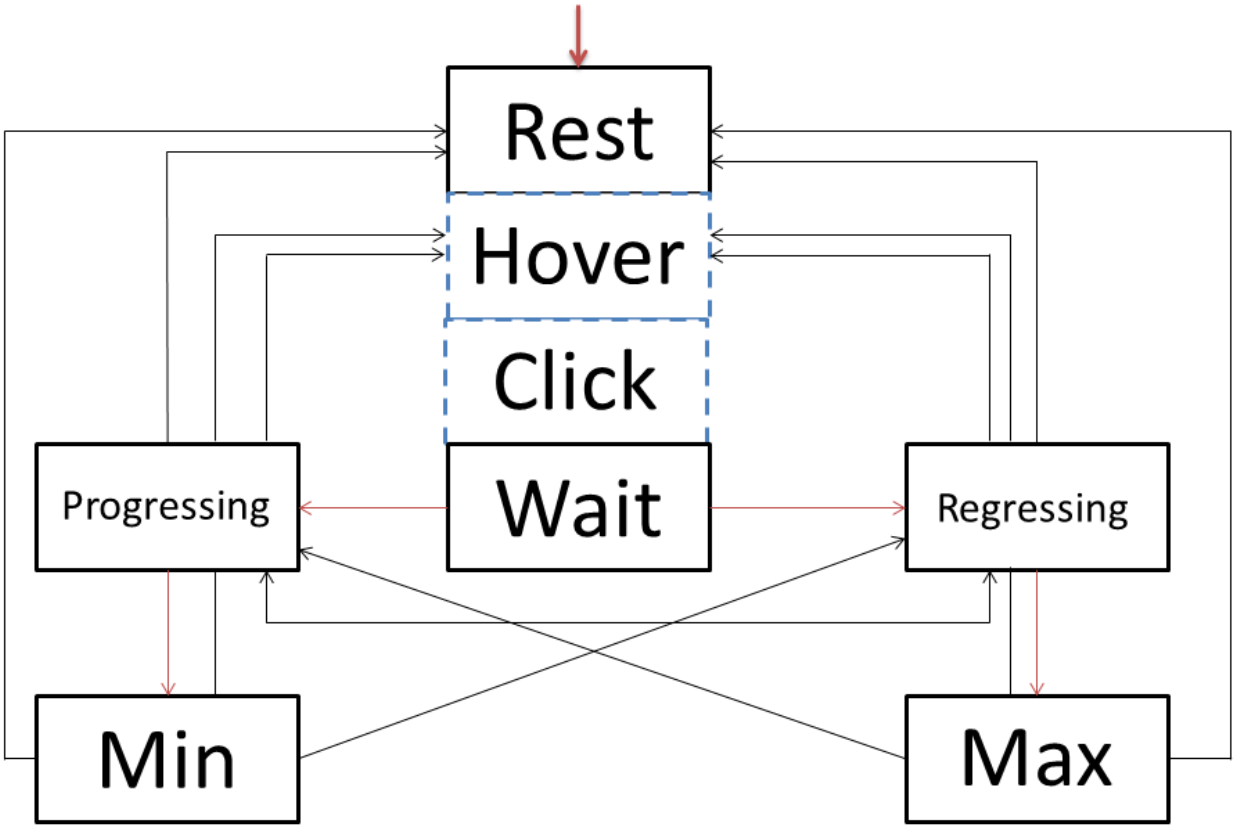
Example of use



Here there is a simple example of use, in which we are modeling something like a button. We are combining «hover» and «click» and adding at the and this X state, that represents the semantic behind the button itself.

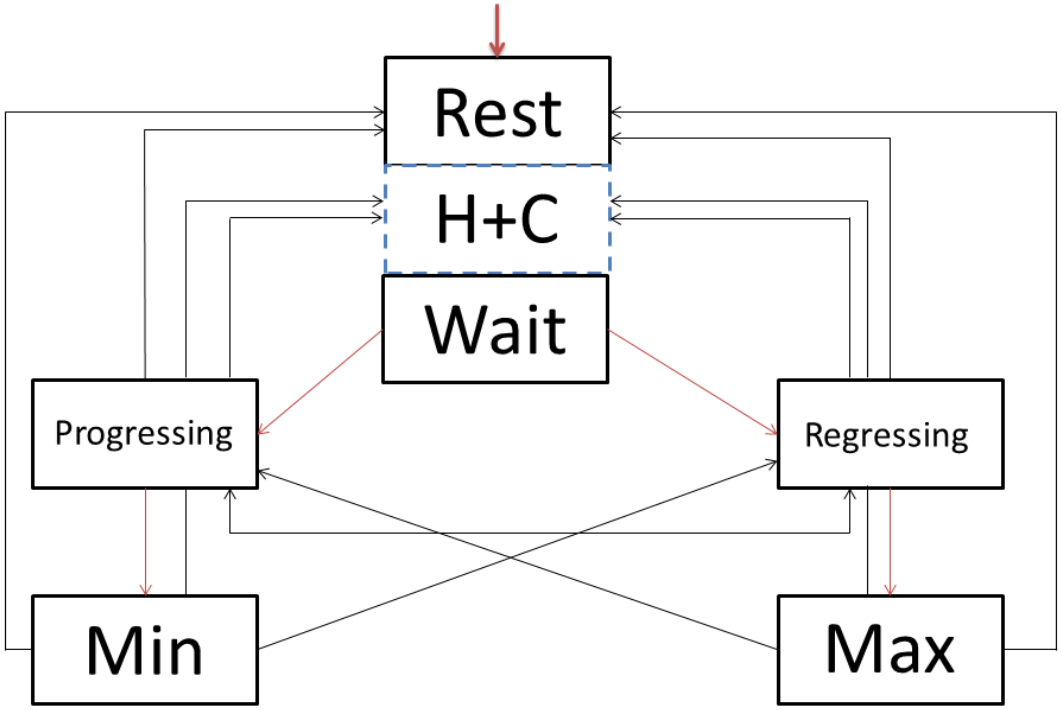
Notice that the **blue dotted boxes** represent our building bricks, while the other components of the FSA represent the semantic of the particular visualization that we are modeling.

Example of use



Here we are modeling, always using a combination of «hover» and «click», one of the handles of a slider. This model could be applied to every handle of the slider, and the final state of the widget is the AND between the current state of each FSA.

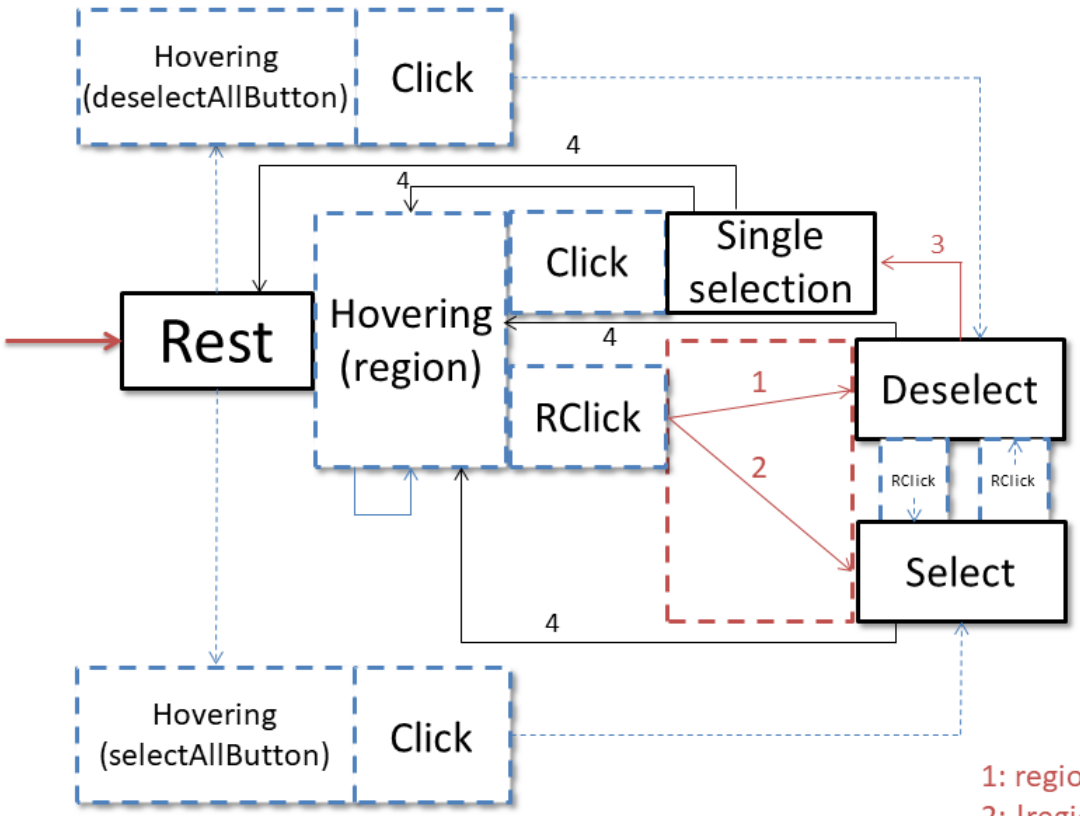
Example of use



This is the same model as the previous one, in which we exploit the building bricks by placing both «hover» and «click» in one widget that contains both, to make the graph more readable.

Notice that transitions represented by red arrows are conditional transitions.

Example of use



Here, as a final test, we are modeling the brexitVisualization, using, whenever we can, our building bricks.

Notice that blue dotted arrows are not transactions, but just a way to visualize where a brick is «attached».

The Rclick boxes between Deselect and Select have arrows inside to visualize the «direction» of the transition inside the brick.

The red dotted box represents the conditional portion of a brick.

The «hovering(region)» brick has a blue arrow going into itself because it can be POST of itself an infinite number of times.

Questions

1. Is this really what we want?
2. Can we really achieve the final goal?
3. Are there any flaws in the theory?
4. Is the notation (visual and not) bad?
5. Are there other atomic widgets?

Thanks for your attention!