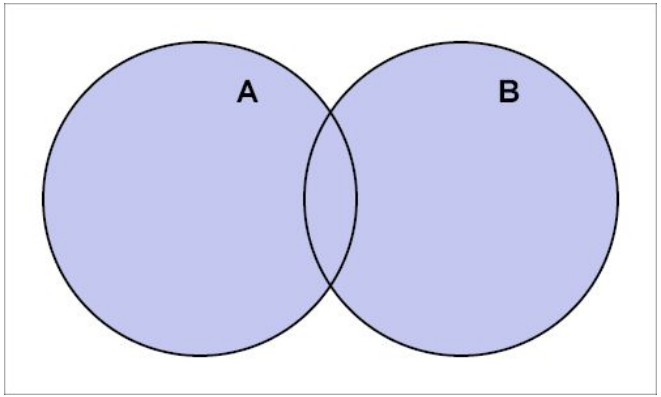
Domain-Specific Modelling of complex User Interfaces

Corrado Ballabio
University of Antwerp - 16/12/16

Statecharts

- perfect for modelling timed discrete-event systems
- not suitable for complex user interfaces*

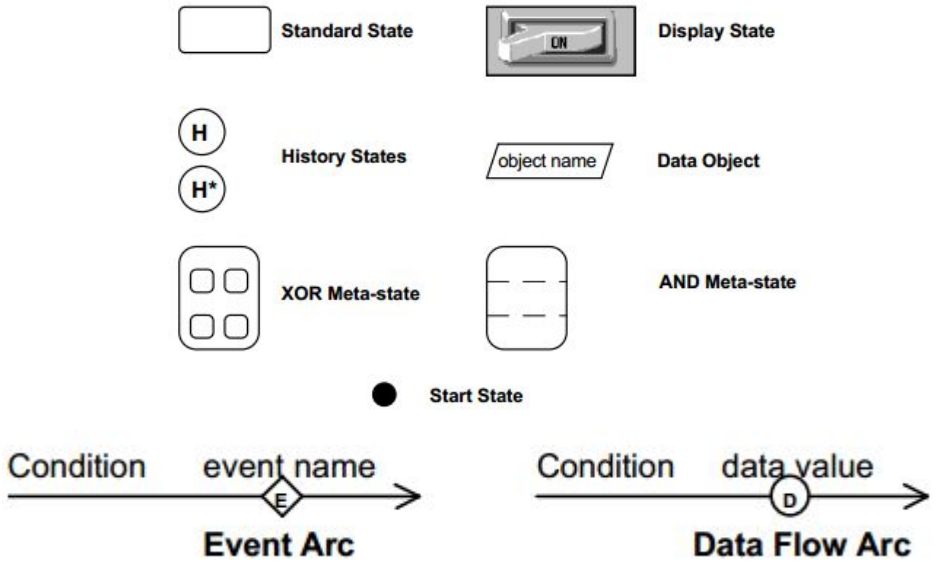


creation of extended formalisms

*:Hans Vangheluwe et al., SCCD: SCXML Extended with Class Diagrams

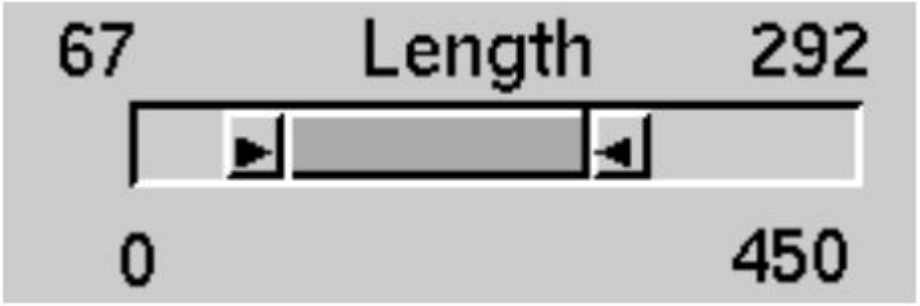
Interactive Object Graph*

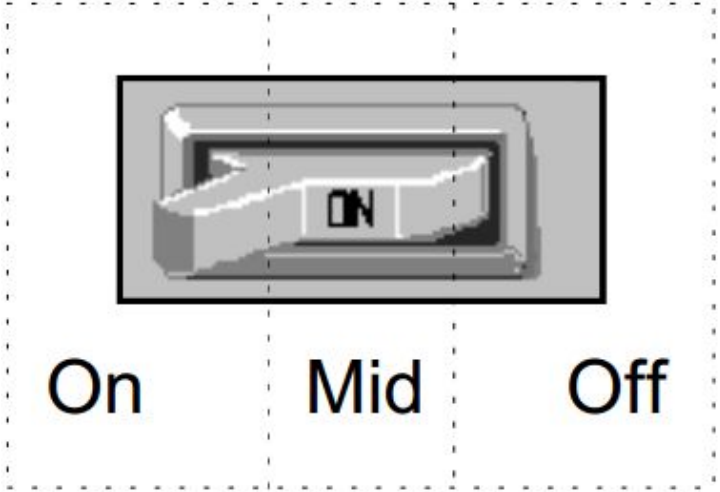
- designing of widgets user interface
- extends Statechart syntax with new nodes and arcs



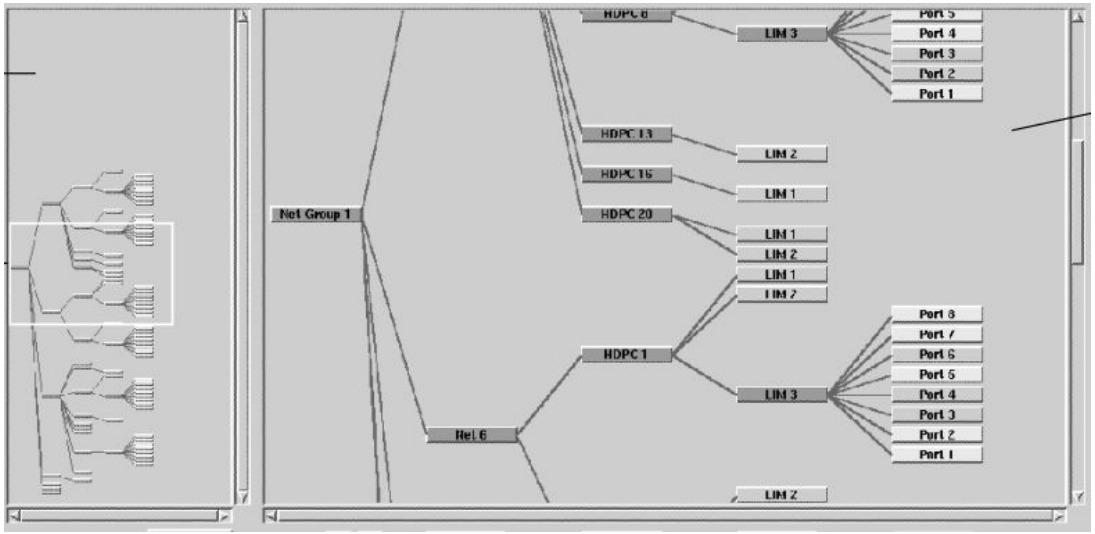
*:David Carr et al.,using interaction object graphs to specify graphical widgets, University of Maryland, 1994

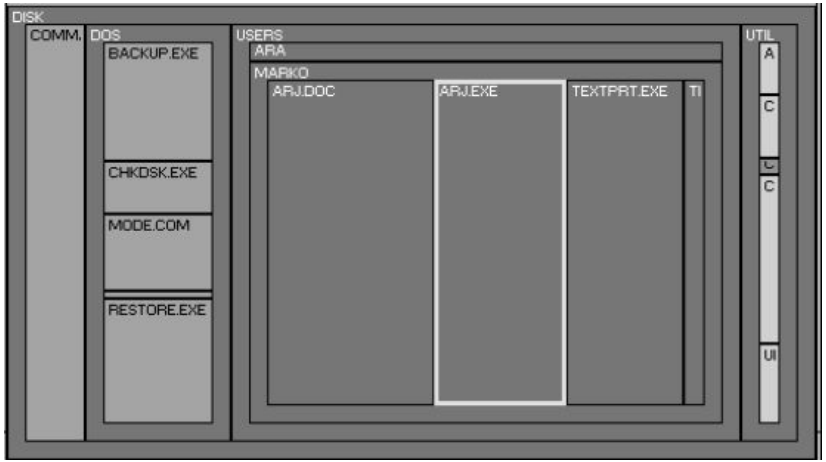
- and defines a new way for describing transitions:
  - **BNS**: booleans, numbers and strings
  - **points**: ordered pair of number
  - **region**: set of points
  - **icon**: region with a graphical representation
  - **view port:** region with a mapping function for underlying application data
  - **window**: groups all the objects in hierarchic levels
  - **user input:** M@, MΔ, ΔM, Mv, Mˆ, in[region], ~[region], [region]~


67
Length
292
0
450


ON
On
Mid
Off

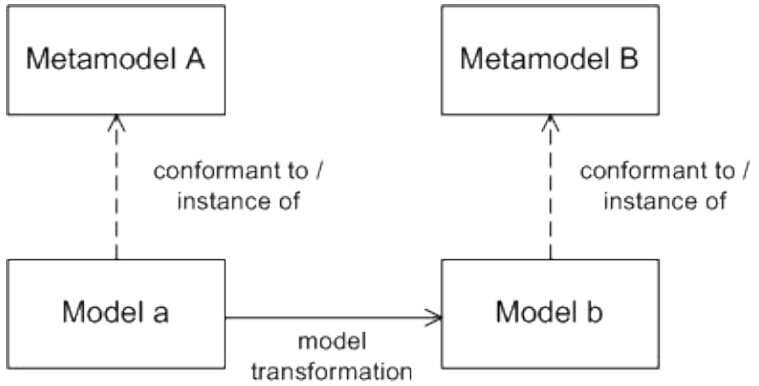

Actress : Azmi, Shabana
A B C D FGH KL M P R S TW Z


HDPC 8
LIM 3
Port 5
Port 4
Port 3
Port 2
Port 1
HDPC 13
LIM 2
HDPC 16
LIM 1
Net Group 1
HDPC 20
LIM 1
LIM 2
LIM 1
LIM 2
Port 8
Port 7
Port 6
Port 5
HDPC 1
LIM 3
Port 4
Port 3
Port 2
Port 1
Net 6
LIM 2


DISK
COMM.
DOS
BACKUP.EXE
CHKDSK.EXE
MODE.COM
RESTORE.EXE
USERS
ARA
MARKO
ARJ.DOC
ARJ.EXE
TEXTPRT.EXE
TI
UTIL
A
C
C
UI

lack of being able to prototype and directly test the specification

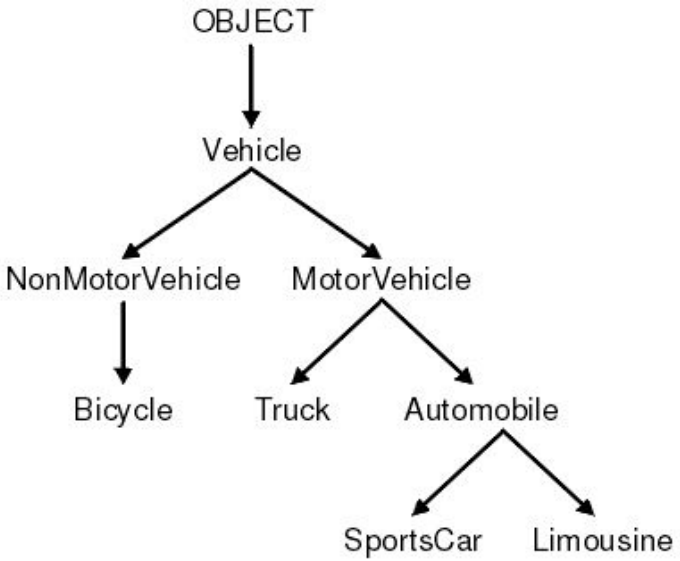
need for a model transformation that allows to draw the widgets and directly execute them



SCCD

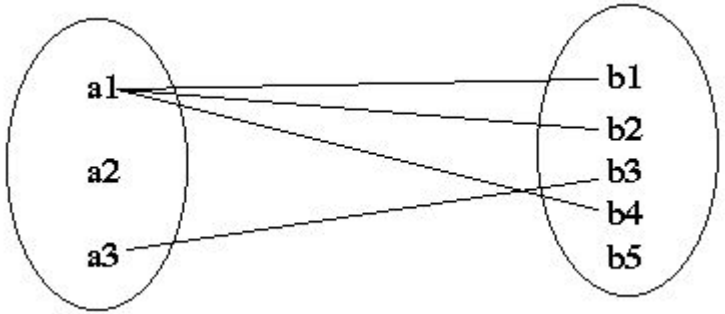
- combines Statechart and Class Diagram Formalisms
- fills the software complexity gap
- adds structural object-oriented expressiveness
- concrete syntax in SCCDXML

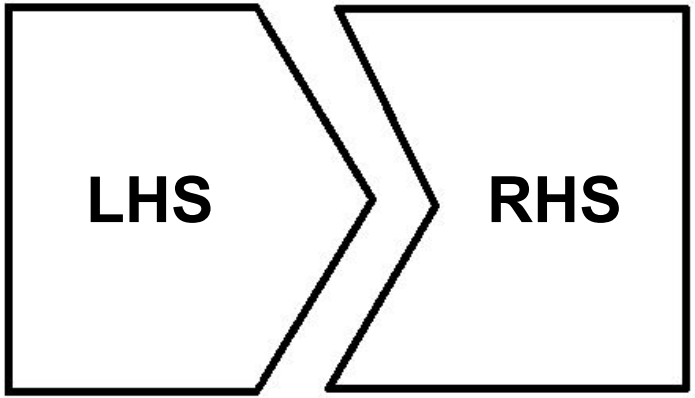
- System structure ➔ Classes
- System behaviour ➔ Statechart



Implementation

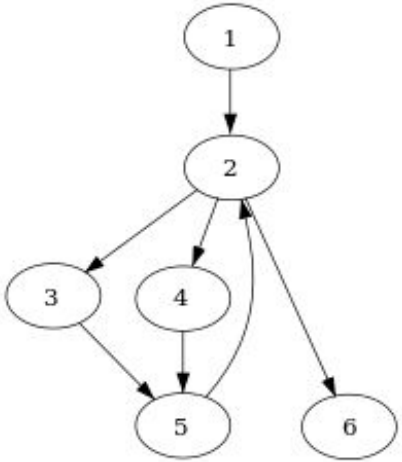
- mapping elements from both the formalisms





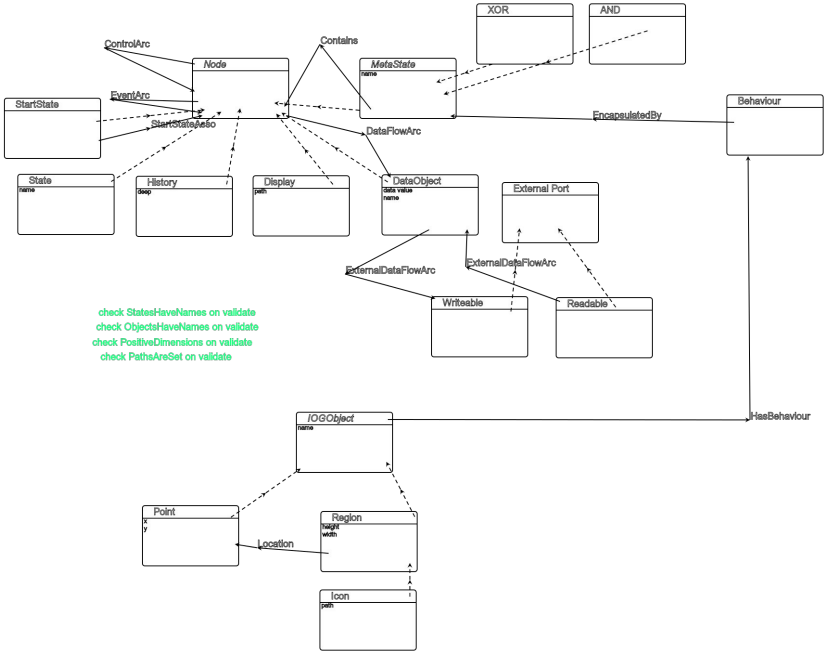
- transformation rules

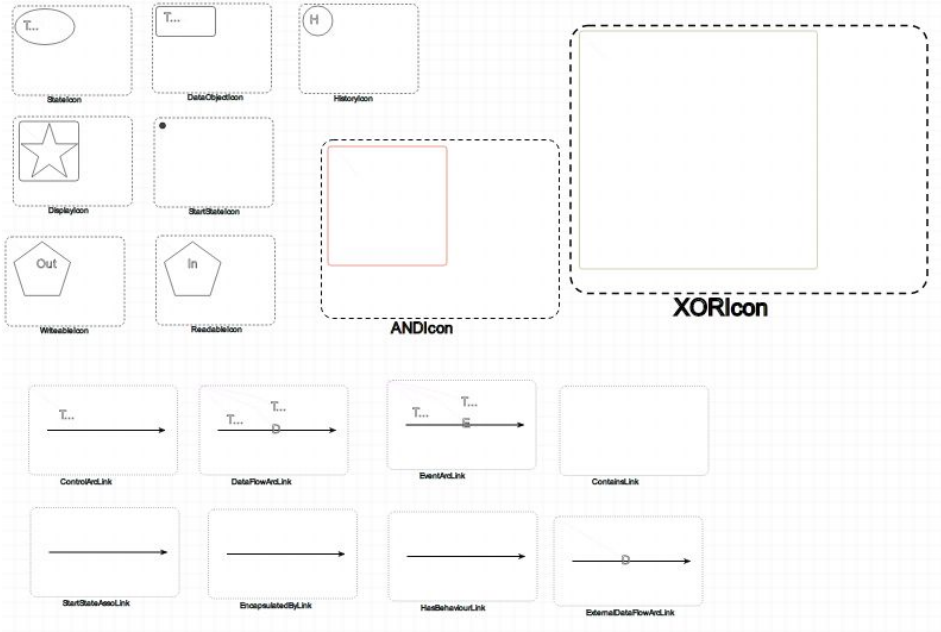
- Scheduling of the rules



- code generation and widgets testing

Previous work

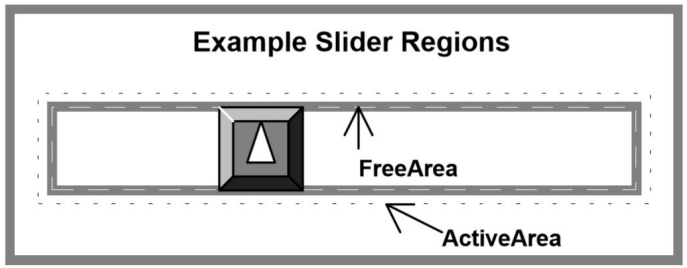


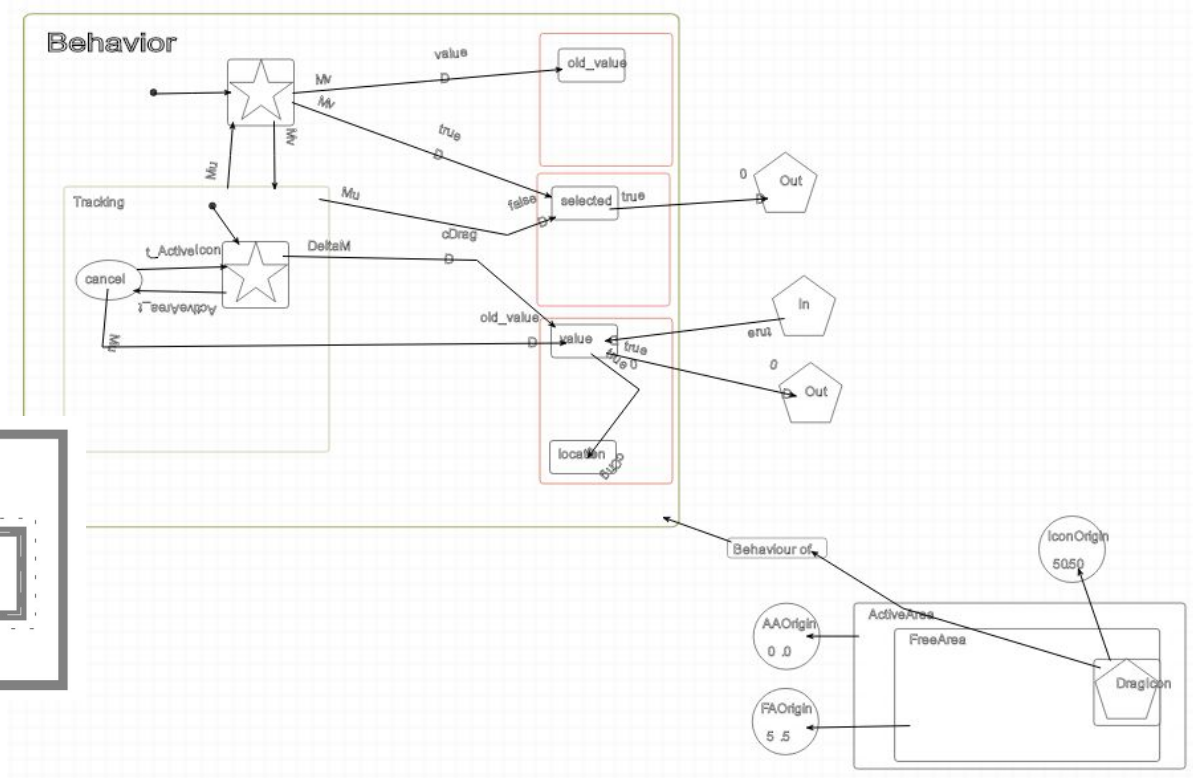


*:Pieter Aerts,Domain-Specific Modelling of complex User Interfaces, University of Antwerp, 2015

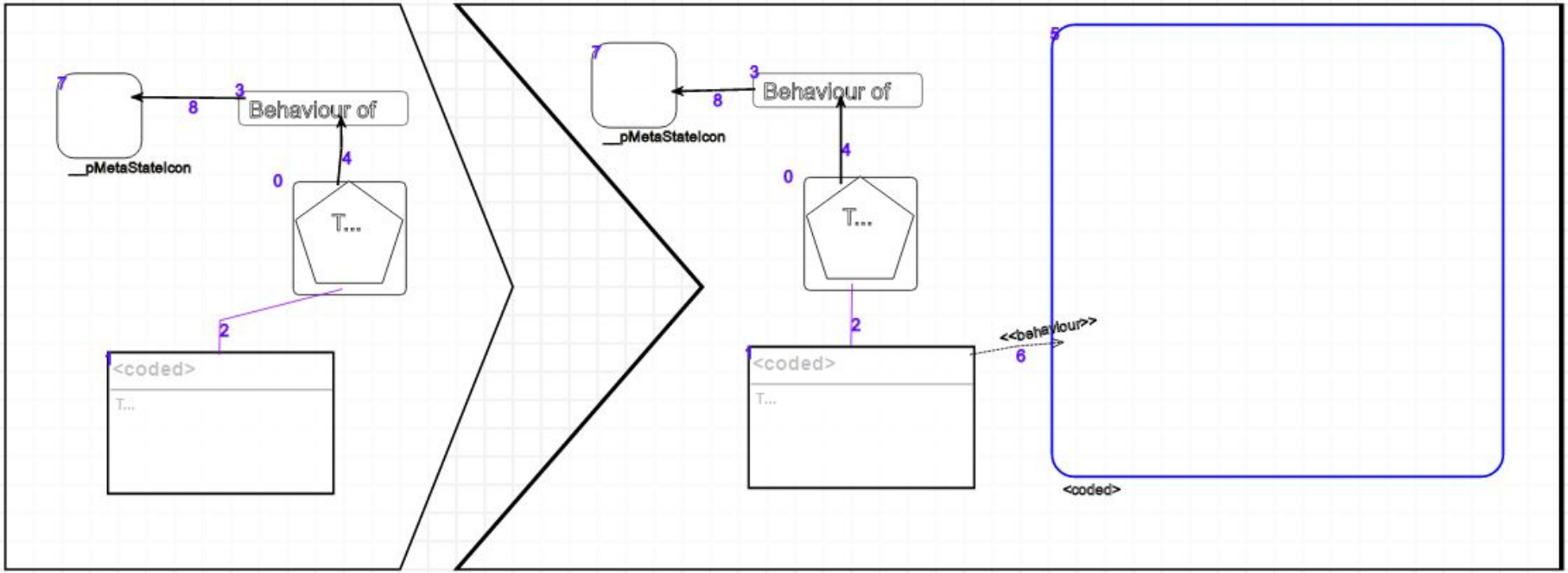
Implementation

let's focus on the implementation of a Drag Icon widget





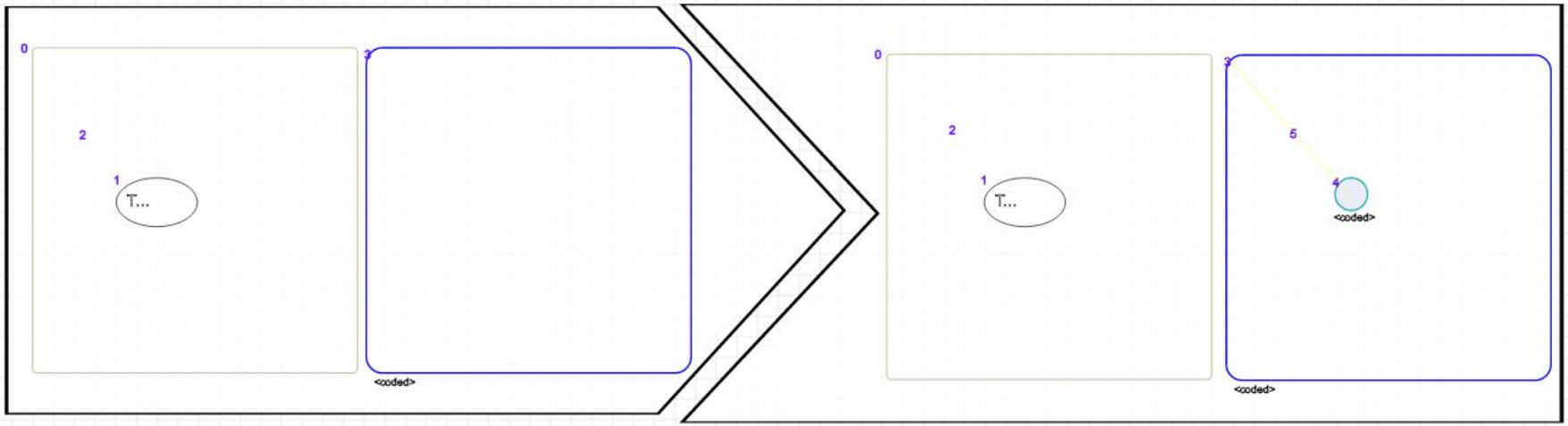
create a starting point



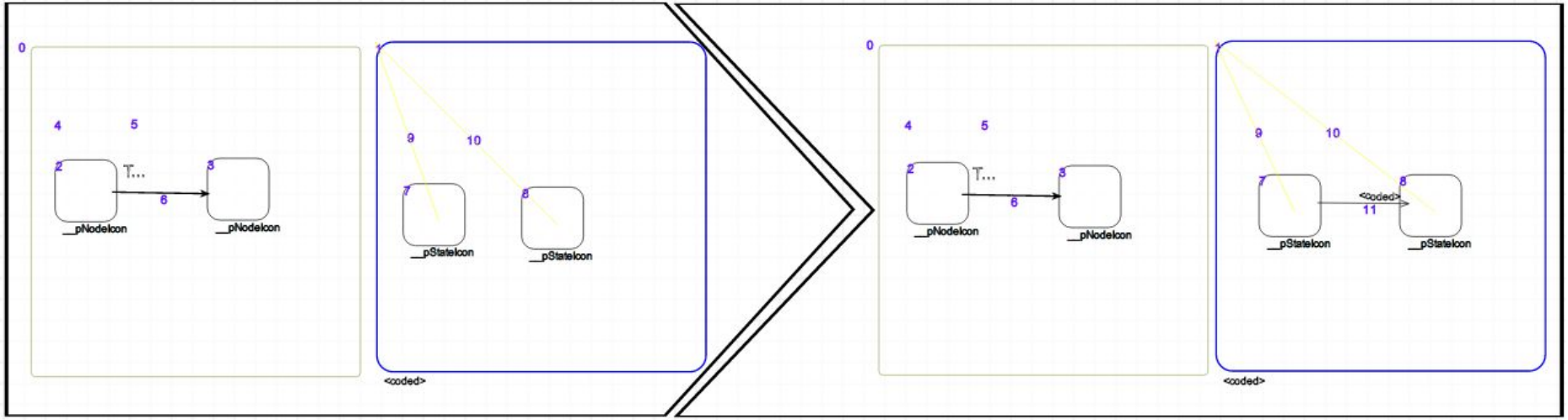
creation of the states

- XOR → composite state
- nodes
- icon
- start → isStart attribute

normal state



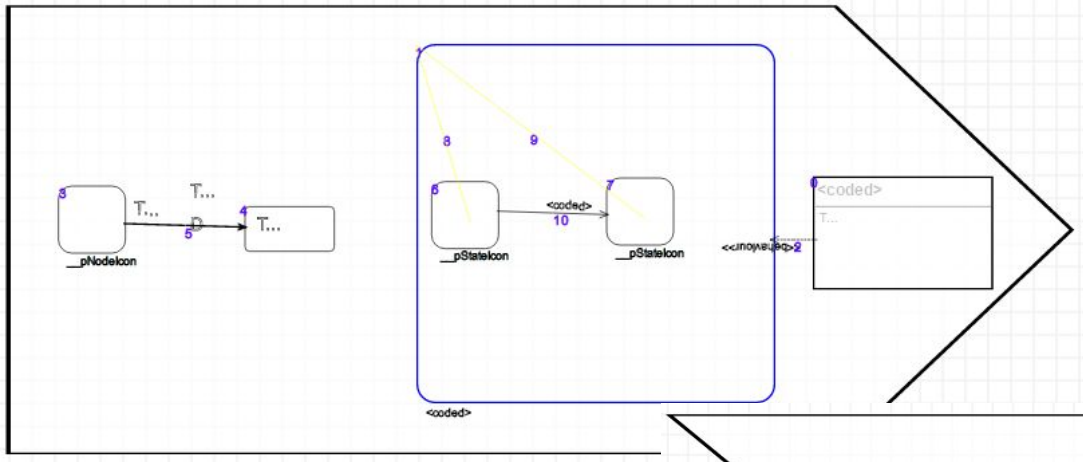
creation of the transition



- arcs between nodes and metastates
- only with control arcs

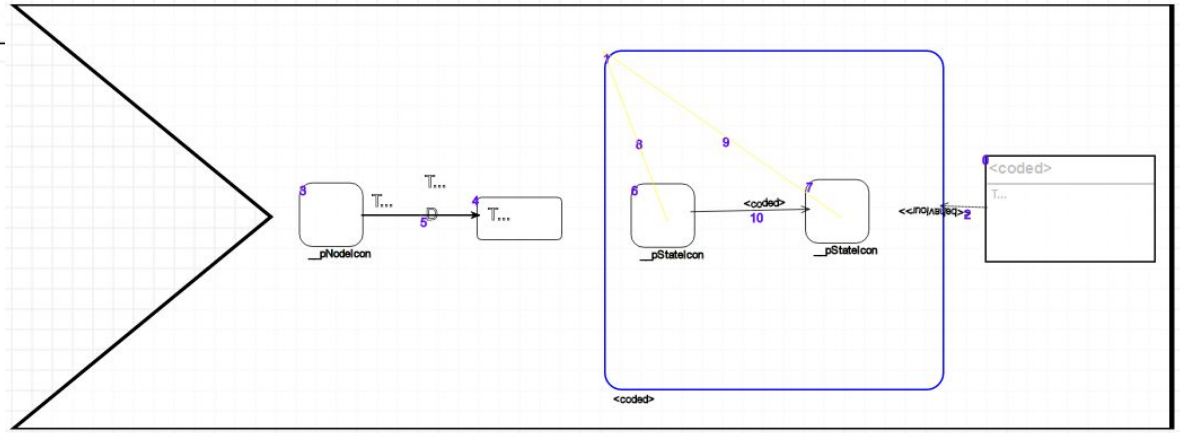
- the new transition has the display and event attribute set as the condition

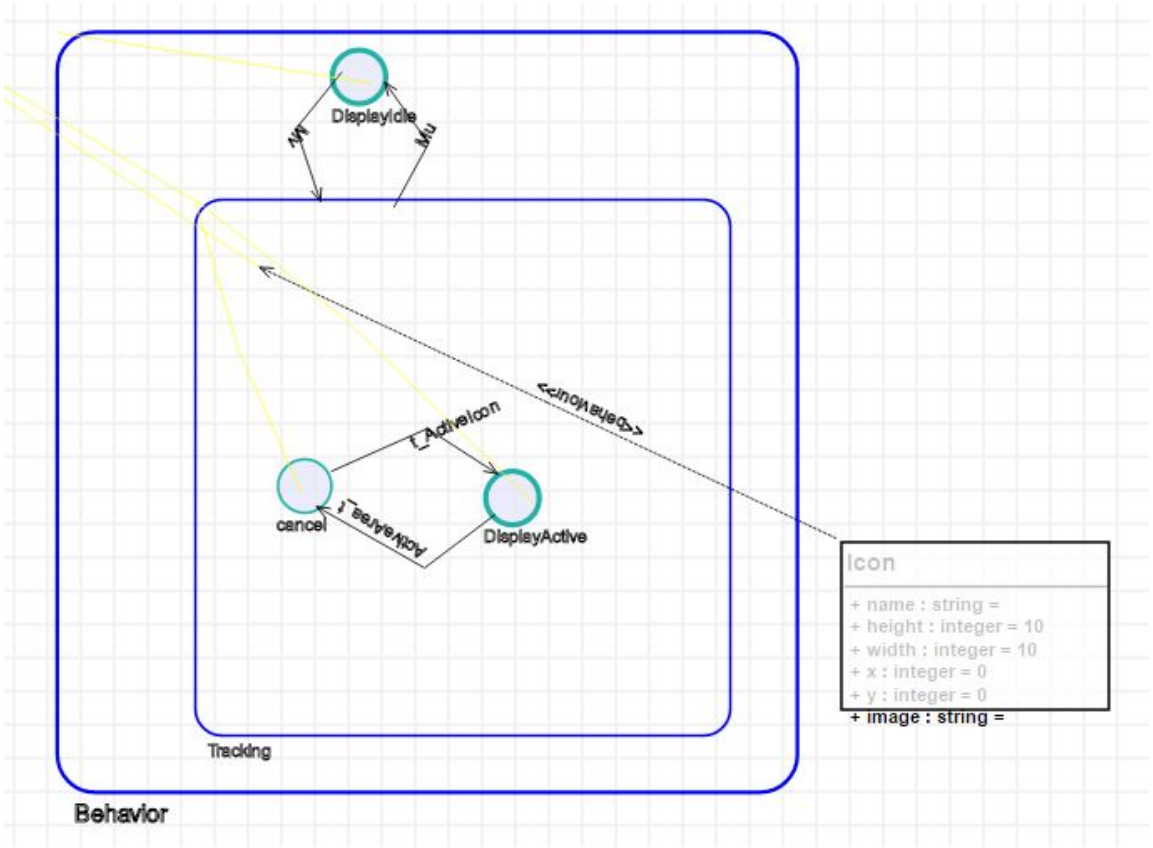
setting the actions



creates inside the class the methods for setting the attribute

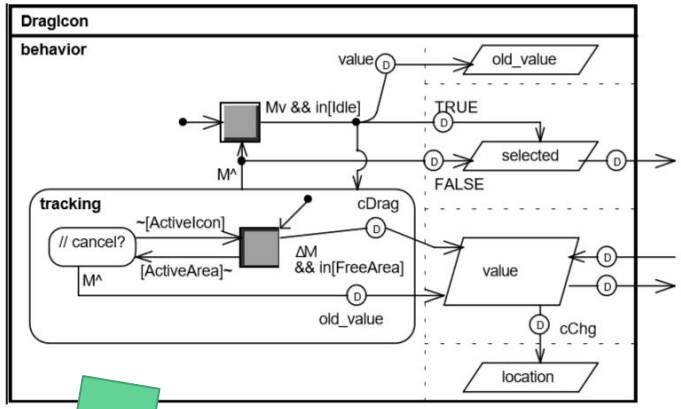
the method is called as action when the transition is triggered

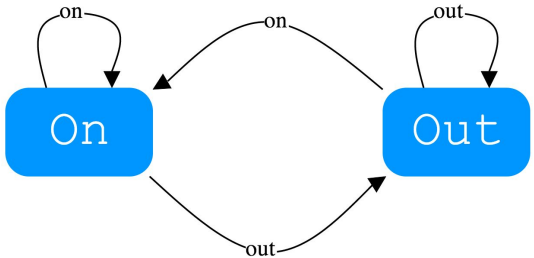




The final result is a SCCD diagram, i.e. a class diagram whose behaviour is encapsulated in a statechart

exporting the model





...and finally