Domain-Specific Modelling of complex User Interfaces

Implementation of the Interaction Object Graph in AToMPM

Pieter Aerts

Universiteit Antwerpen

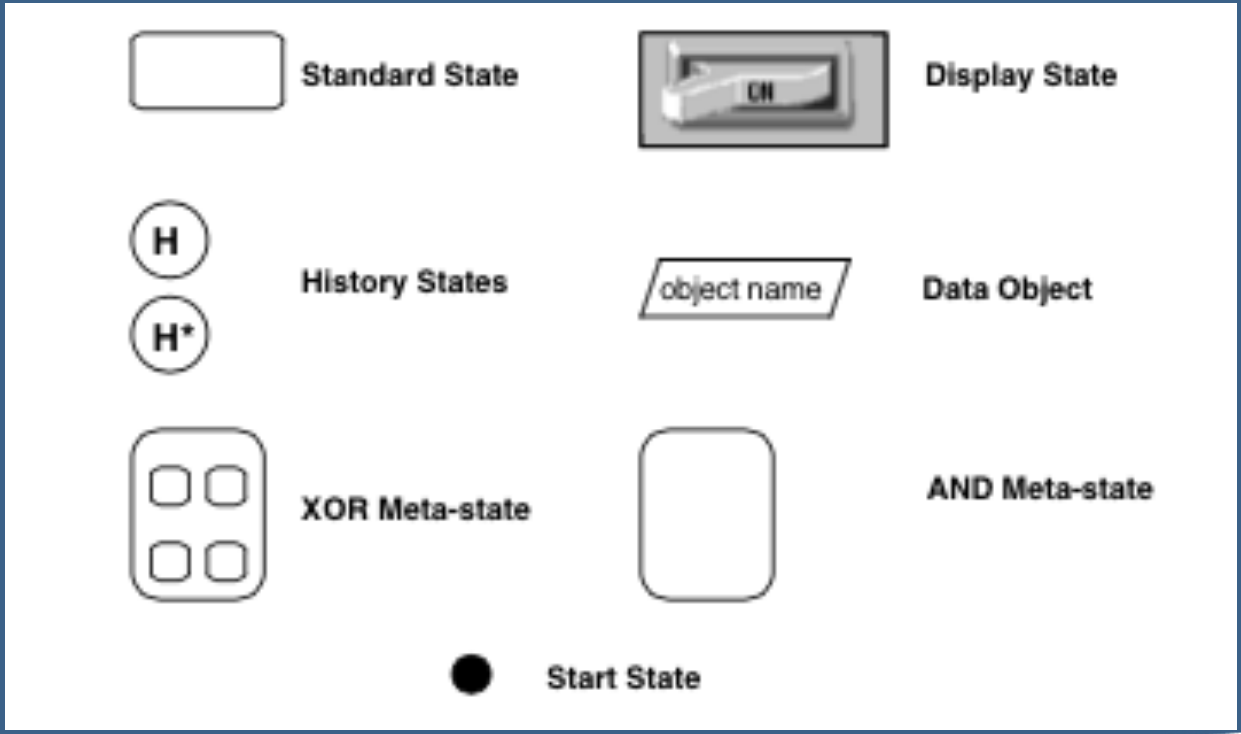
Table of contents

1. Interaction Object Graph

- Graphical widget specification
- Based on:
  - Interface Representation Graphs
    - Data flow
    - Constraint specifications
  - Statecharts
    - Transition based execution model
    - Meta-states
    - History states

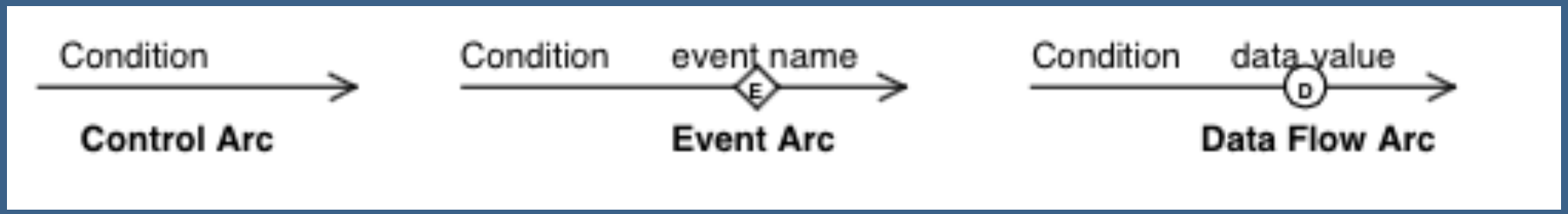
1. Interaction Object Graph

- Node symbols



1. Interaction Object Graph

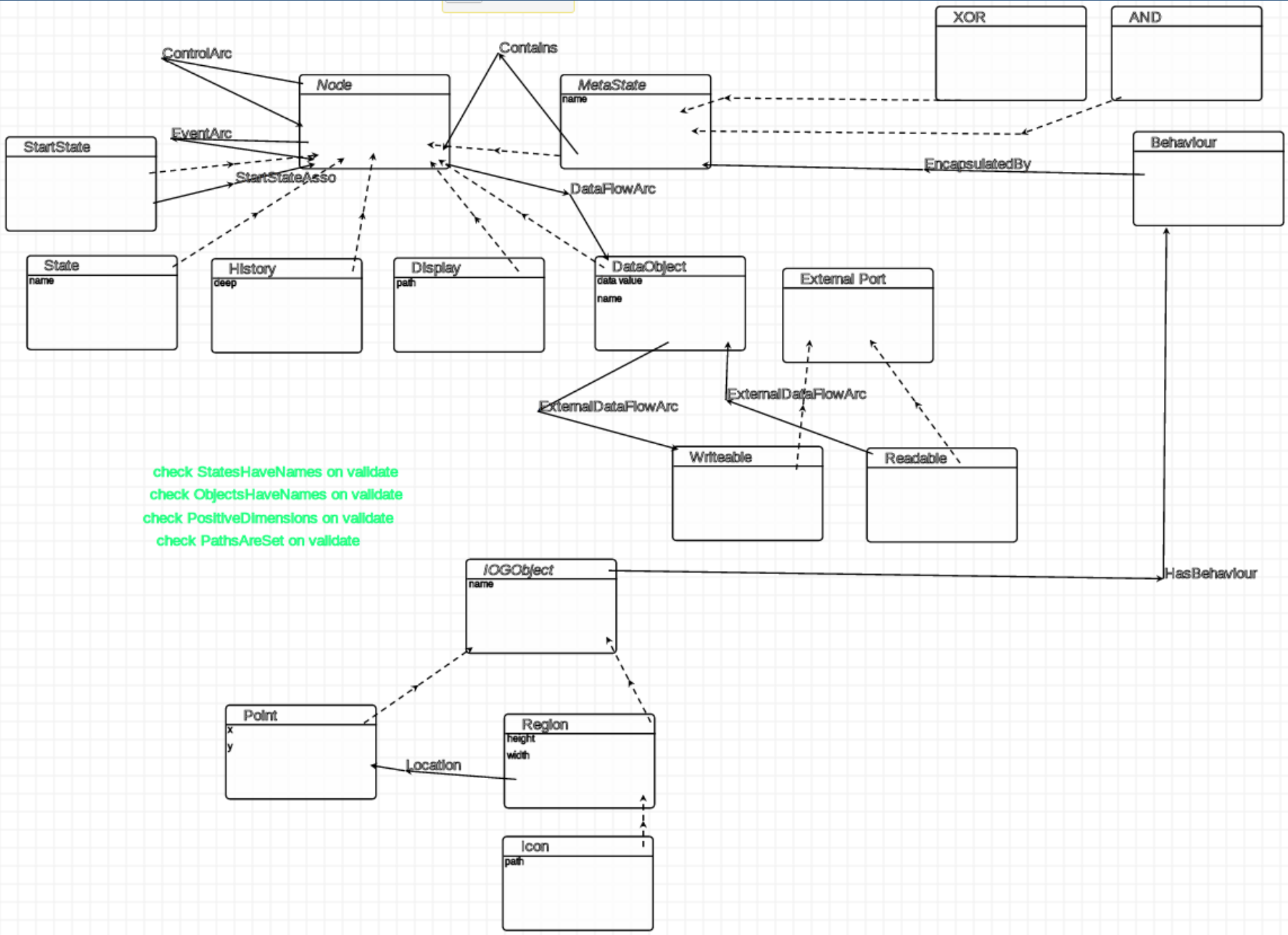
- Arc symbols



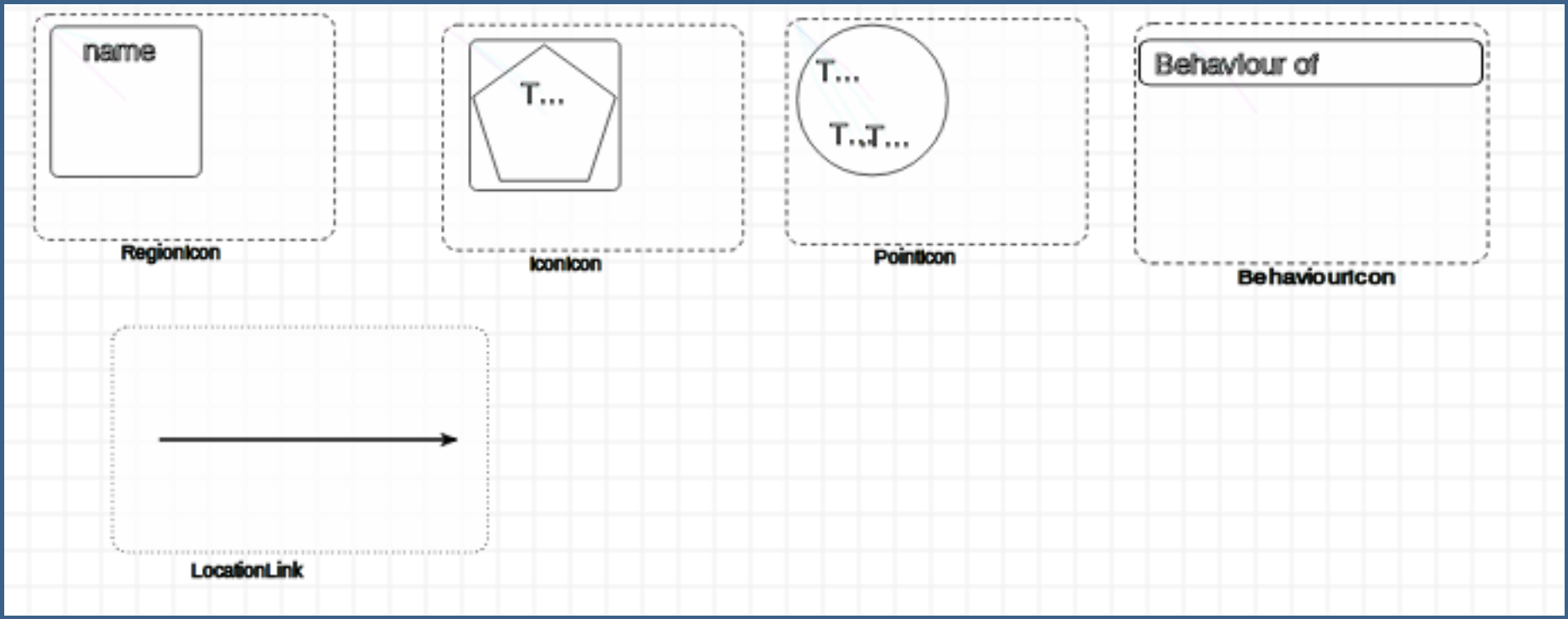
1. Interaction Object Graph

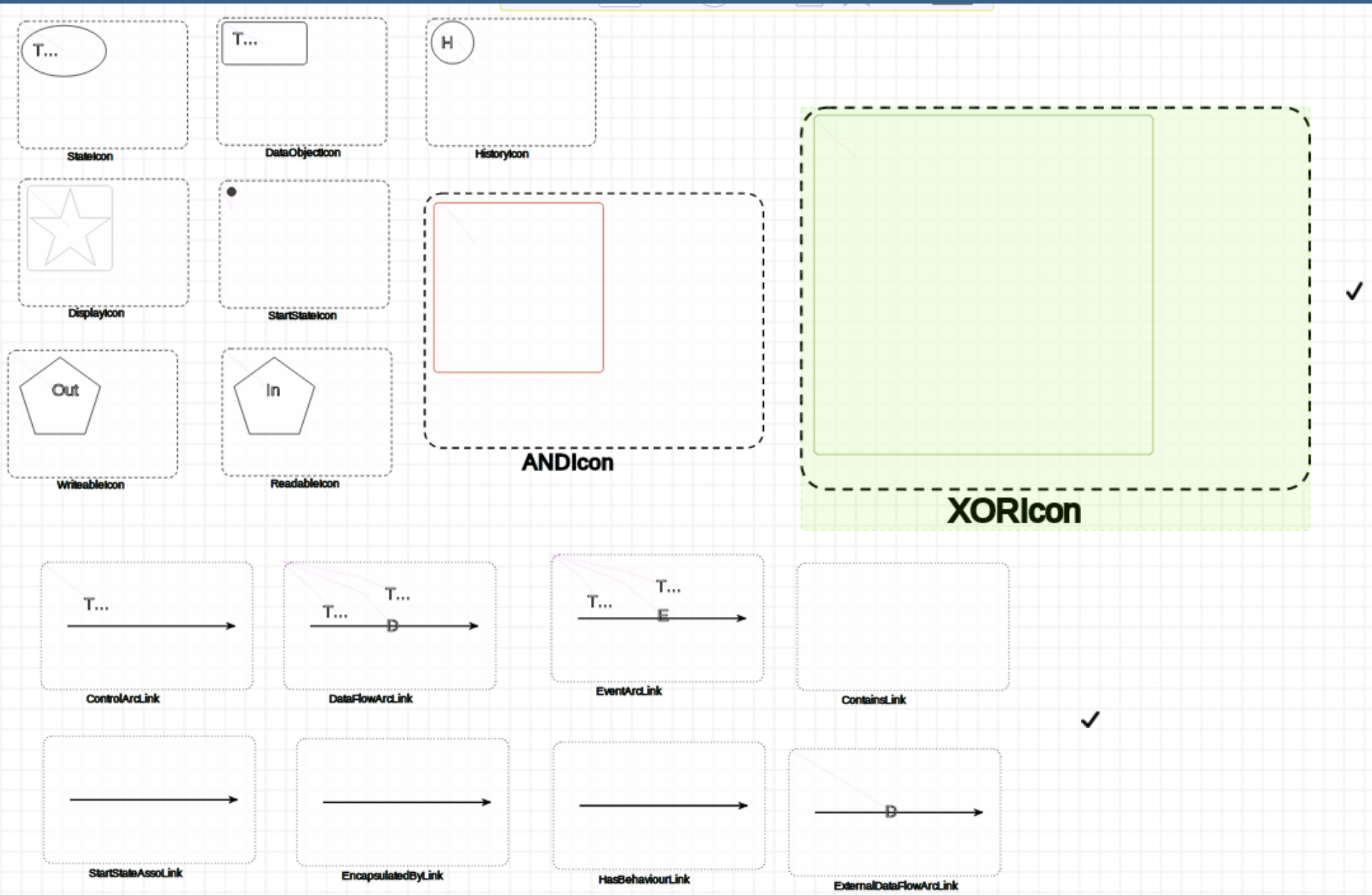
- Object Types
  - Booleans
  - Numbers
    - Real / integer
  - Strings
  - Points
    - (x,y)
  - Regions
    - Origin, height, width
  - Icons
    - Region with graphical display
  - Window
  - User inputs

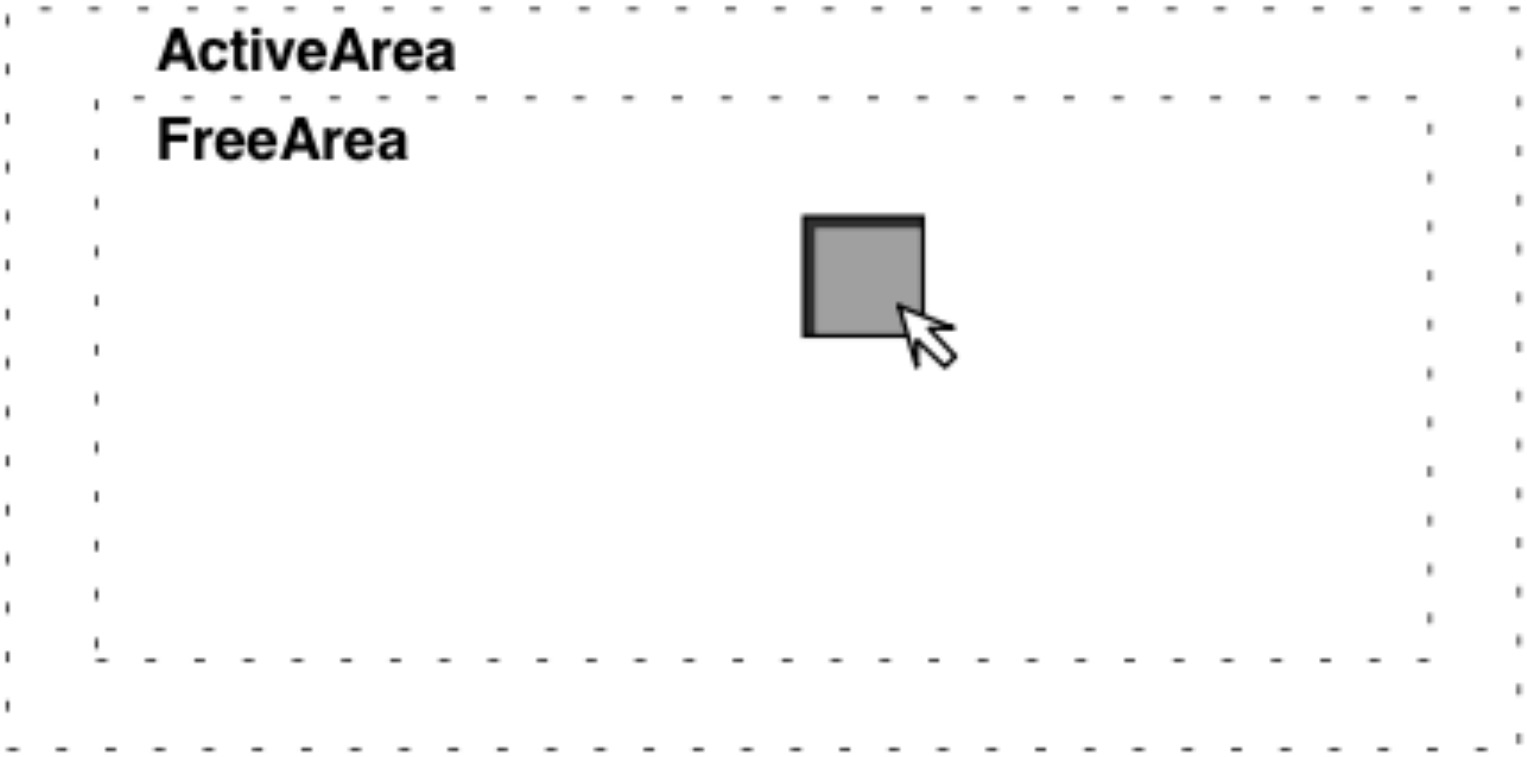
2. AToMPM: Metamodel

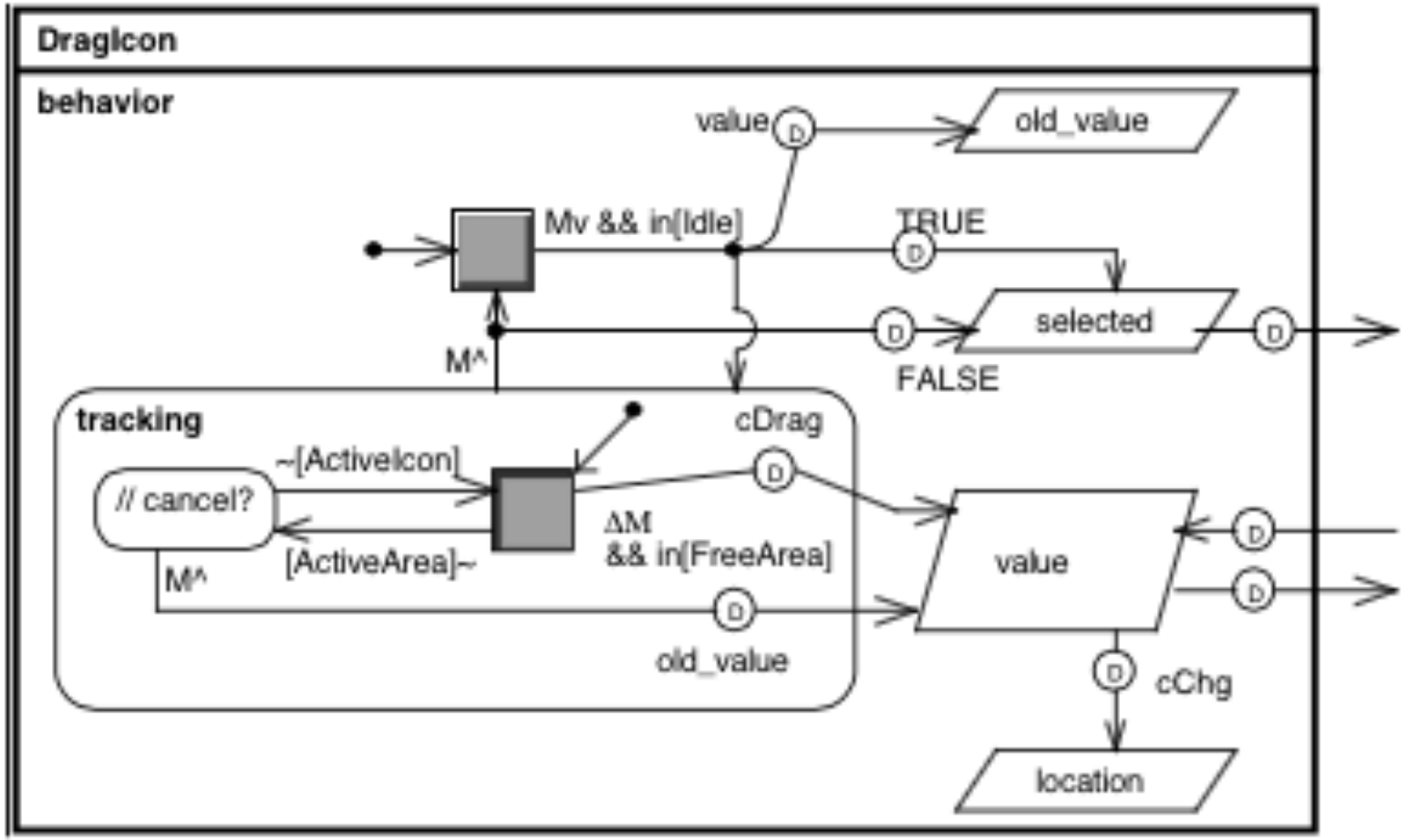

XOR
AND
ControlArc
Contains
Node
MetaState
name
EventArc
StartState
StartStateAsso
DataFlowArc
Behaviour
EncapsulatedBy
State
name
History
deep
Display
path
DataObject
data value
name
External Port
ExternalDataFlowArc
ExternalDataFlowArc
Writeable
Readable
check StatesHaveNames on validate
check ObjectsHaveNames on validate
check PositiveDimensions on validate
check PathsAreSet on validate
IOGObject
name
HasBehaviour
Point
x
y
Location
Region
height
width
Icon
path

2. AToMPM: Concrete syntax

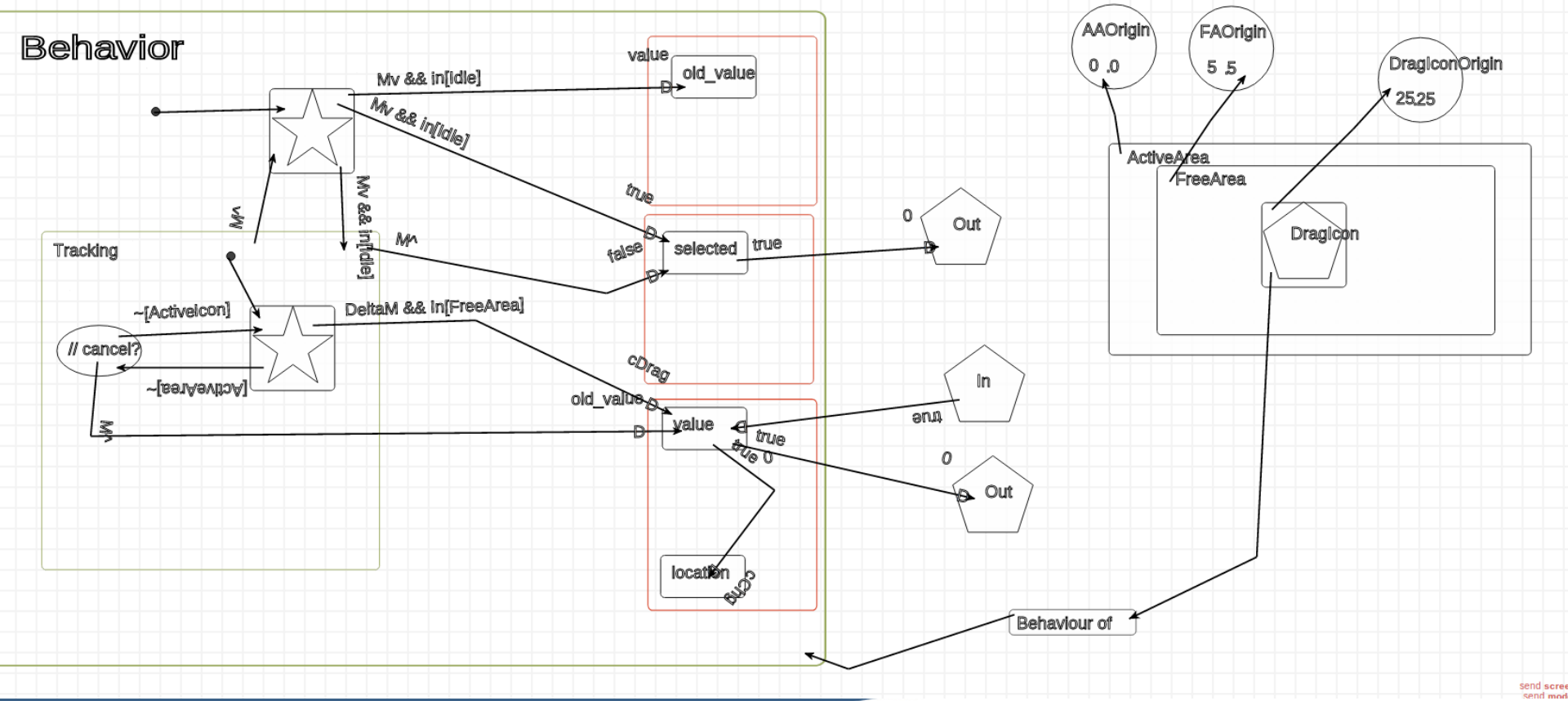


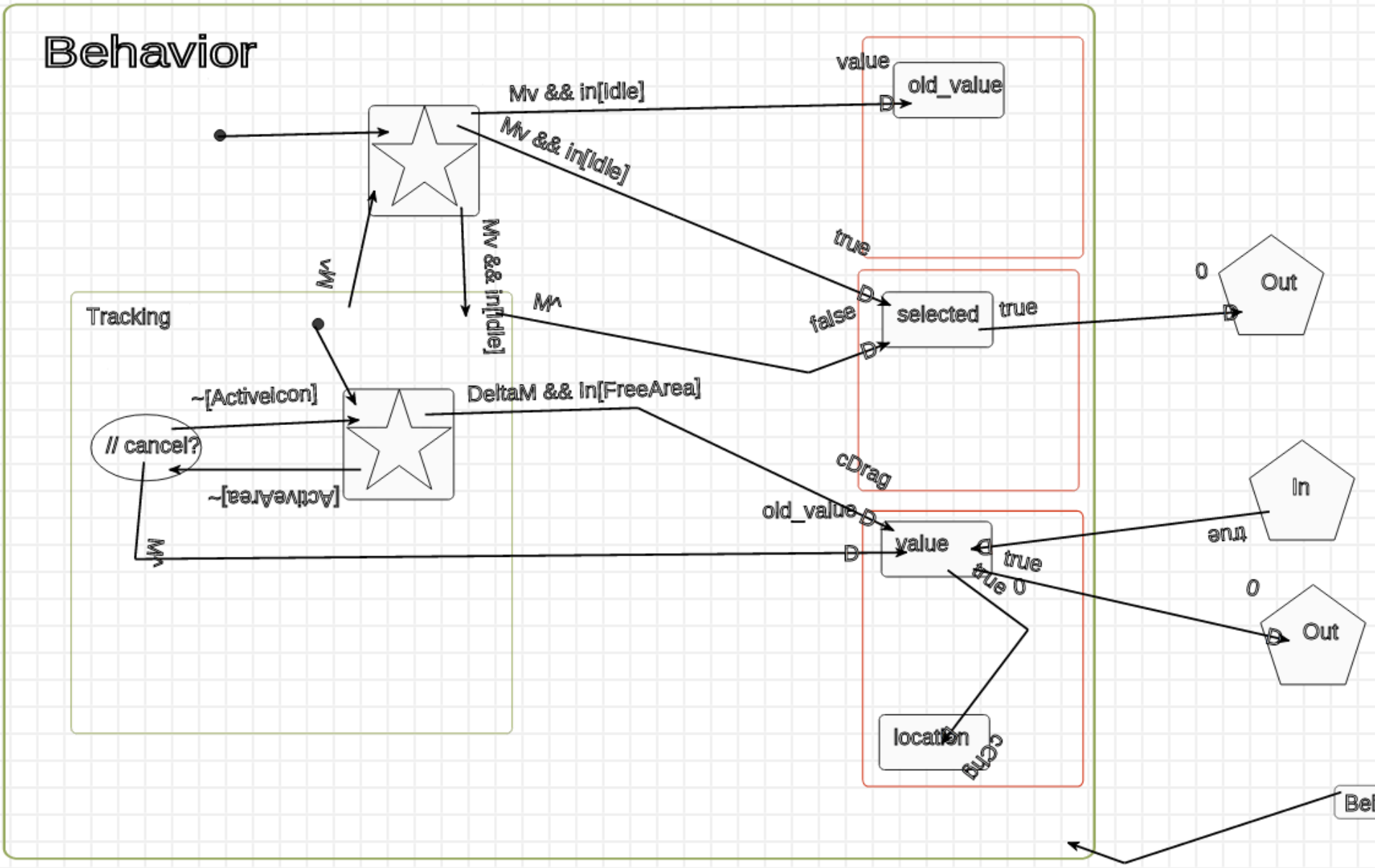

T...
StateIcon
T...
DataObjectIcon
H
HistoryIcon
DisplayIcon
StartStateIcon
Out
WriteableIcon
In
ReadableIcon
ANDIcon
XORIcon
T...
ControlArcLink
T...
T...
D
DataFlowArcLink
T...
T...
E
EventArcLink
ContainsLink
StartStateAssoLink
EncapsulatedByLink
HasBehaviourLink
D
ExternalDataFlowArcLink


ActiveArea
FreeArea

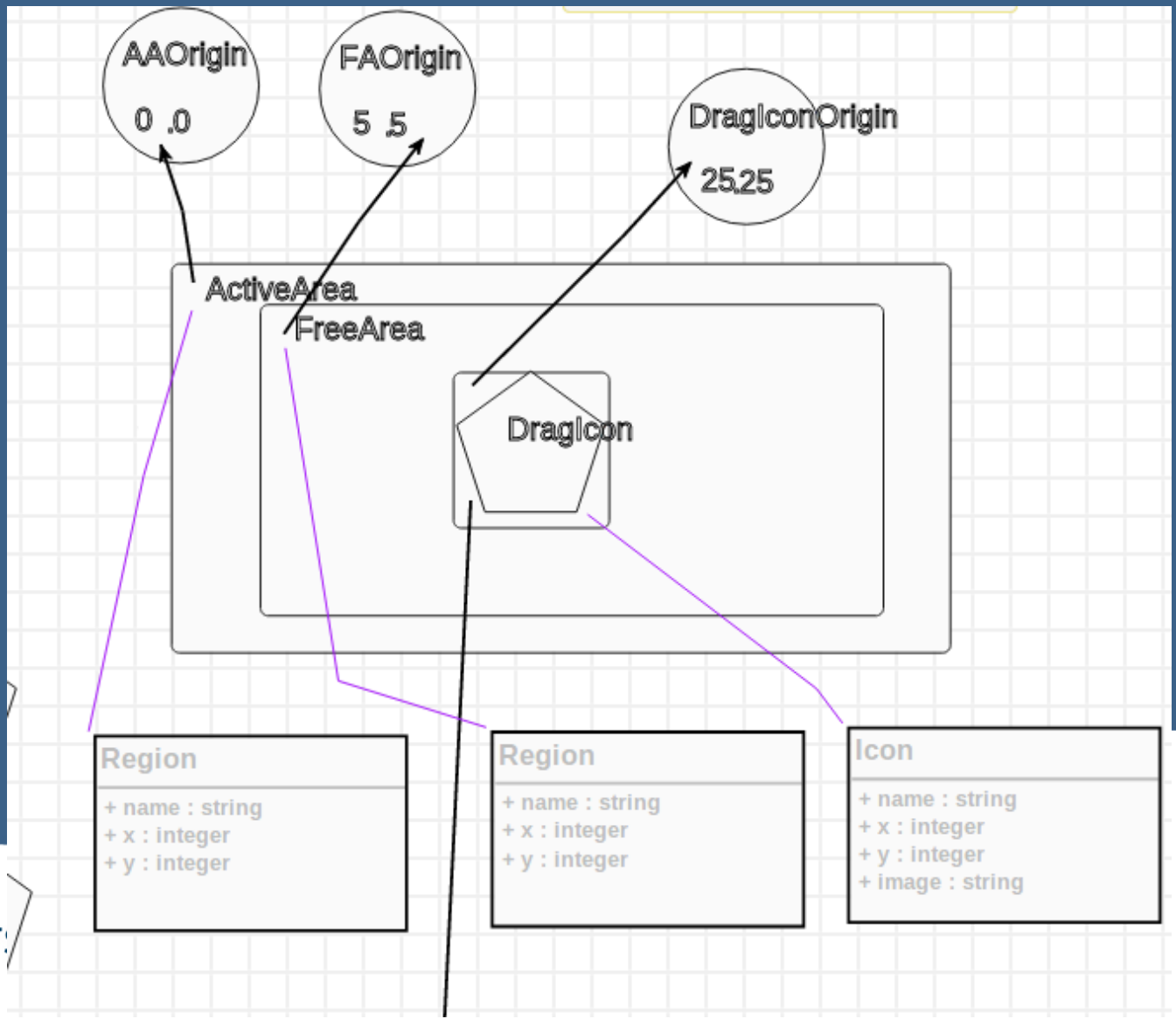

DragIcon
behavior
value
old_value
Mv && in[Idle]
TRUE
selected
M^
FALSE
tracking
cDrag
~[ActiveIcon]
// cancel?
[ActiveArea]~
ΔM
&& in[FreeArea]
M^
value
old_value
cChg
location

2. DraggableIcon model




Behavior
Mv && in[idle]
Mv && in[idle]
value
old_value
true
Mv
Mv && in[idle]
Mv
Tracking
false
selected
true
0
Out
~[ActiveIcon]
DeltaM && in[FreeArea]
// cancel?
~[ActiveArea]
cDrag
old_value
value
true
true 0
In
true
0
Out
location
cChg
Mv

3. Mapping onto CD

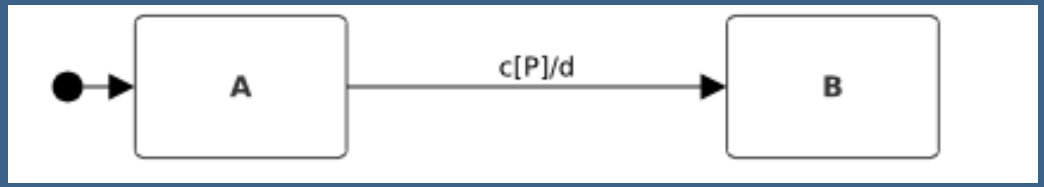


3. Mapping onto SCCD

1. Corresponding constructs
   - Basic states
   - History states
   - Composite and orthogonal states
   - Control arc

2. No corresponding constructs
   - Data Objects
   - Display states
   - Event and data arcs

3. Mapping onto SCCD



- Data Object:
  - add attribute to class
- Display state
  - regular state
  - d = method call
- Event arc
  - Firing condition - > trigger / guard
  - d = event
- Data arc
  - Read / Write - > d = get(), set()

- c = trigger event
- P = guard condition
- d = action / event

4. Conclusion and future work

- Abstract and visual syntax of IOG in AToMPM
  - Creation of valid specification models
- Model transformation to SCCD
- Future work
  - Generating code
    - GUI library
    - Simulate widget
  - Tool for specification and execution

References

- Interaction Object Graphs to Specify and Develop Graphical Widgets
  - David Carr , Ninad Jog , Harsha Kumar, Marko Teittinen and Christopher Ahlberg (1994)

- Statecharts and Class Diagram XML: A general-purpose textual modelling formalism
  - Glenn De Jonghe (2014)