



Xtext + Sirius : <3

Cédric Brun <cedric.brun@obeo.fr>

Who am I ?

CTO @ Obeo

Involved with Eclipse since 2006, Modeling PMC, Councils..

Strategic Member of the Eclipse Foundation



EcoreTools 2.x,

The Modeling Package,

Extended Editing Framework,

ATL

...



Focused on **domain specific modeling** technologies
50 people working on products and customer projects
Services : training, consulting and Open Innovation
Tool Vendor



**Obeo
SmartEA**



**Obeo
Designer**





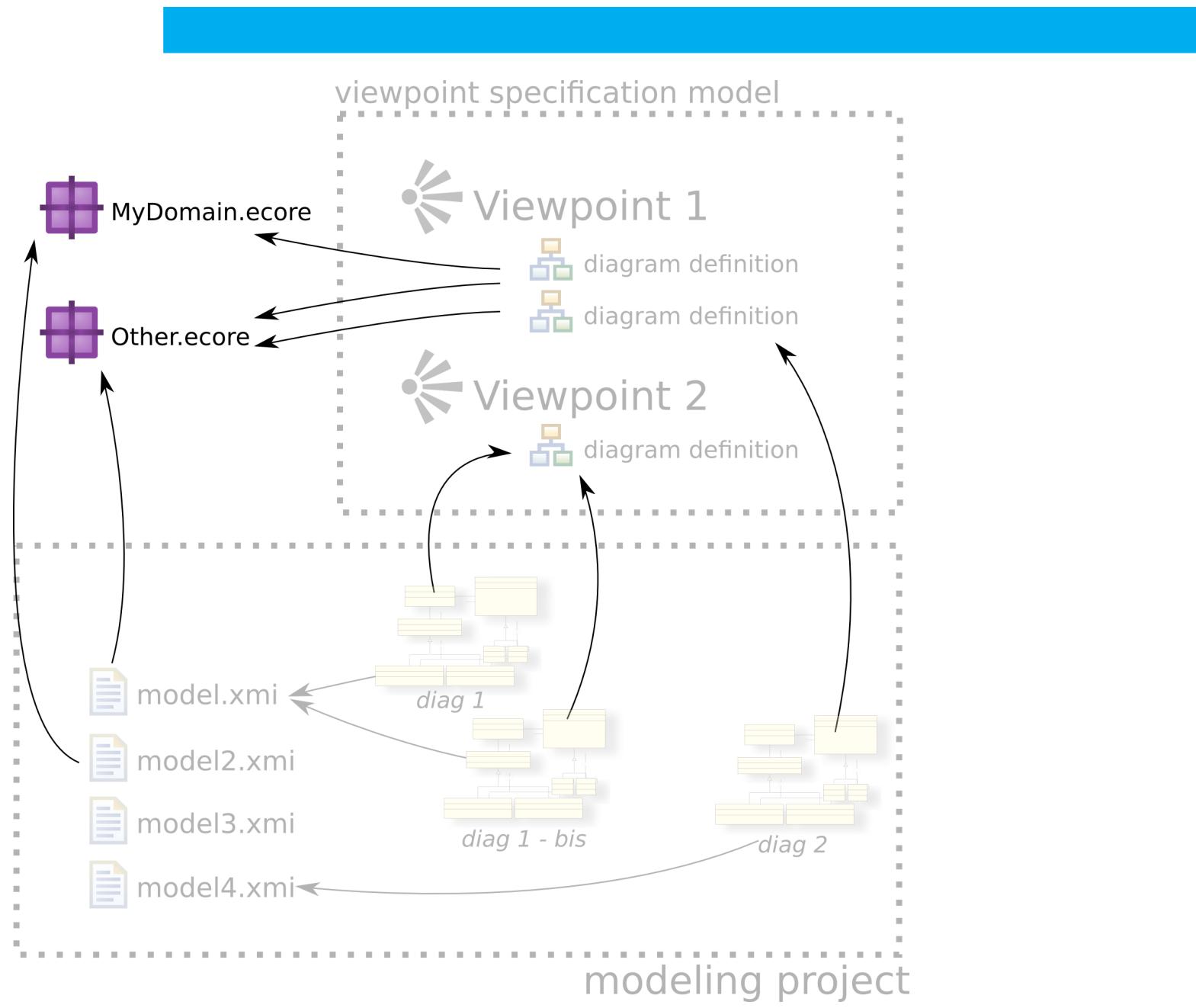
Eclipse Sirius

*« Sirius aims at providing specific multi-view workbenches through **diagram**, **table** or **tree** modeling editors.*

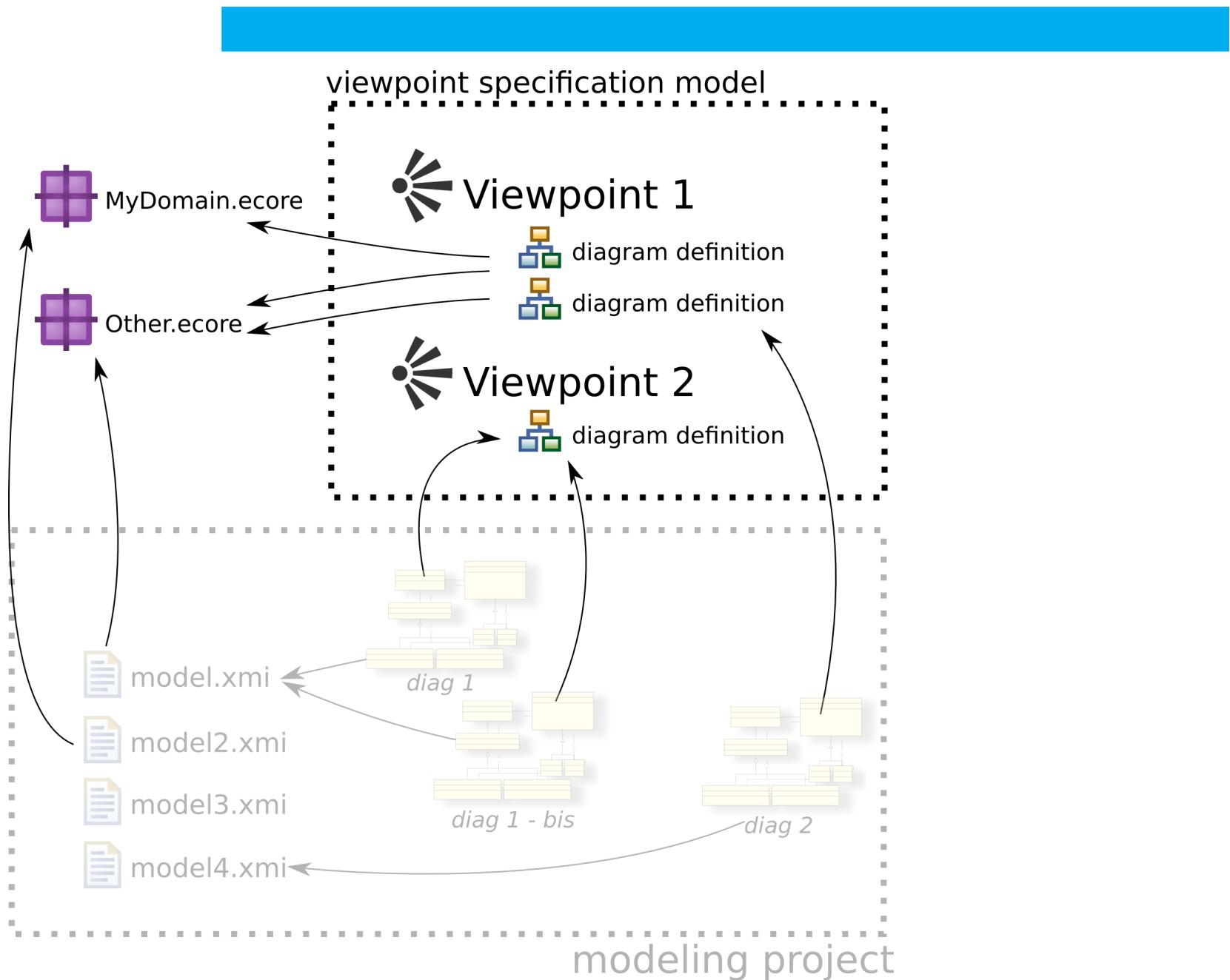
*Users can easily define **their own modeling workbench**, even with very little technical knowledge of Eclipse, while still being able to deeply customize it when needed. »*



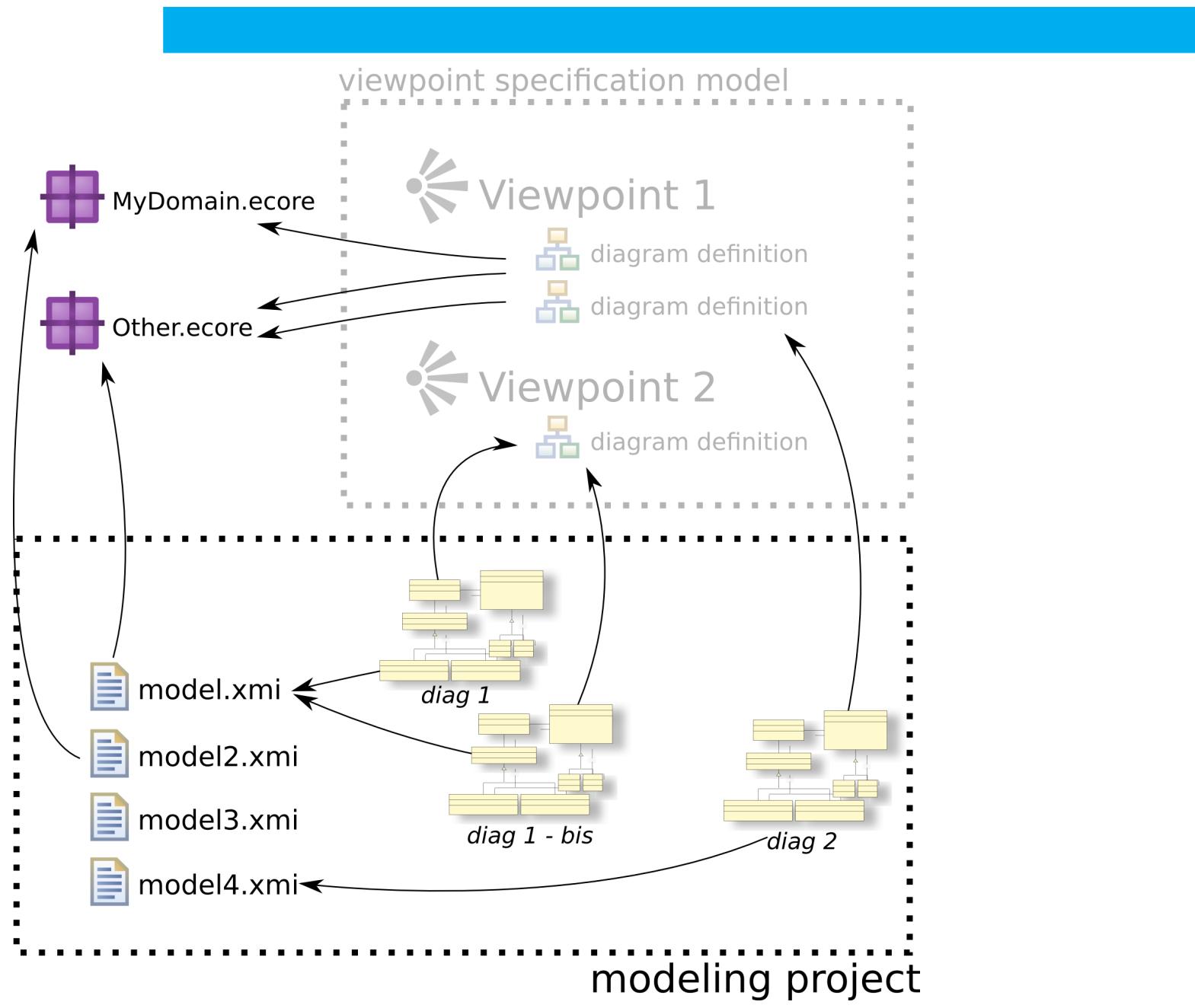
Sirius Principles



Sirius Principles



Sirius Principles



Textual + Graphical ?



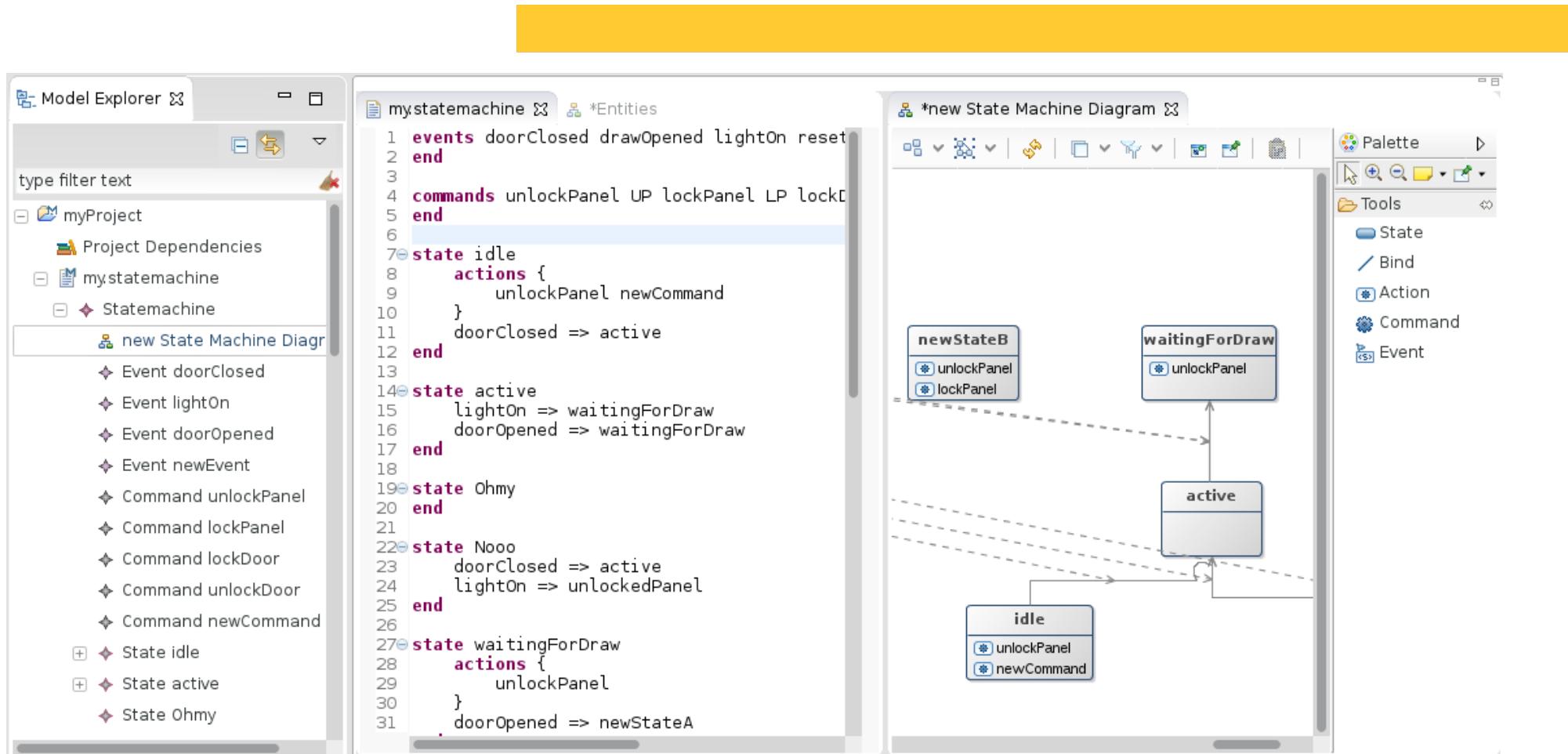
Text is better:

- at operations and sequence
- when combining (expressions)
- ...
- for some users

Diagram is better :

- relationships
- analyzing a design
- ...
- for some users

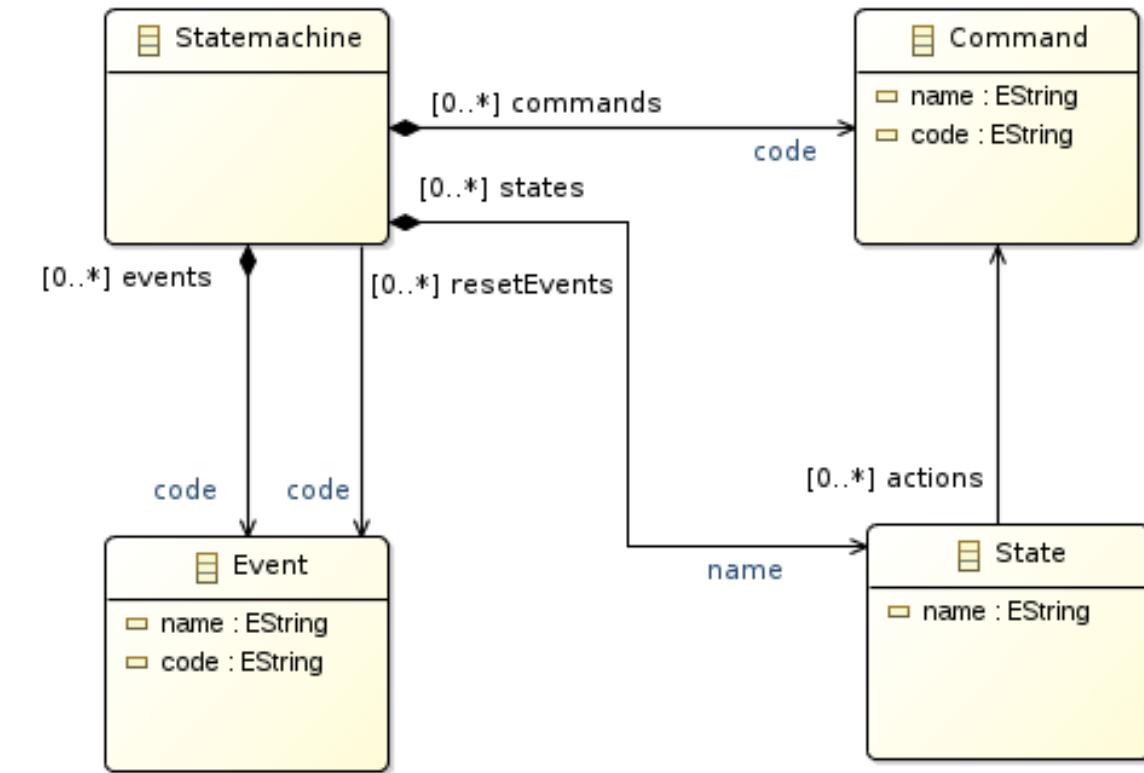
Two views of the same model



File/Workspace level integration
DSL syntax used as the serialization format
DSL AST used in the Sirius Editor

The StateMachine Demo

Xtext 2.6.0
Sirius 1.0.0 RC1
Eclipse Luna



<https://github.com/ObeoNetwork/Xtext-Sirius-integration>

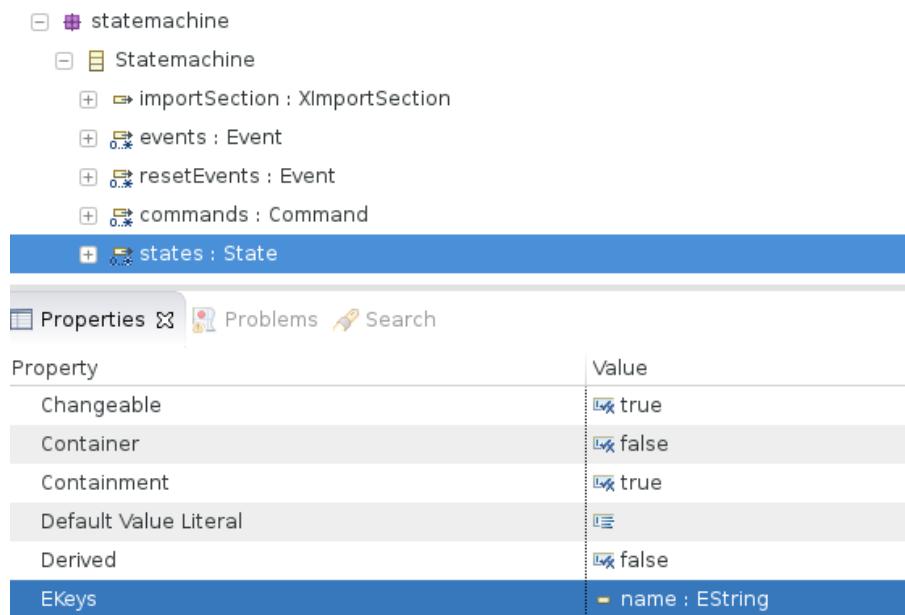
How To ?

☰ ☐ ■■■ Sirius

☐  Sirius Core Runtime	1.0.0.201405260918
☐  Sirius Documentation	1.0.0.201405260918
☐  Sirius IDE Support for Acceleo	1.0.0.201405260918
☐  Sirius Integration with EEF	1.0.0.201405260918
☐  Sirius Integration with XText	1.0.0.201405260918
☐  Sirius Runtime IDE	1.0.0.201405260918
☐  Sirius Runtime Support for Acceleo 3	1.0.0.201405260918
☐  Sirius Runtime Support for OCL	1.0.0.201405260918
☐  Sirius Samples	1.0.0.201405260918
☐  Sirius Specification Editor Support for Acceleo	1.0.0.201405260918
☐  Sirius Specification Environment	1.0.0.201405260918

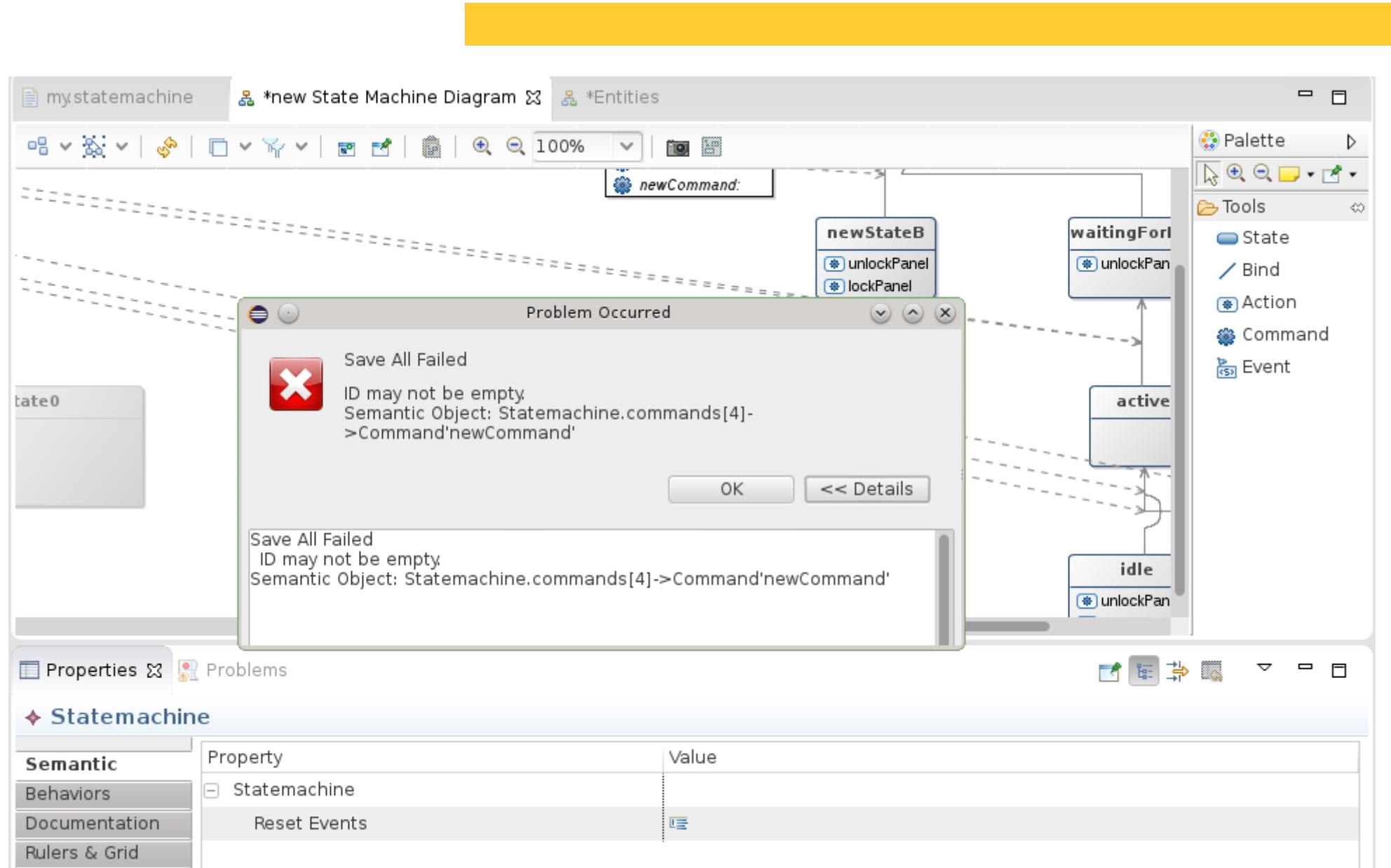
Caveats : the referencing

`href="my.statemachine//@states.4"`



`href="my.statemachine//@states[name='active']"`

Caveats : Am I serializable ?



Caveats : Am I serializable ?



Provide graphical feedback ASAP

Tools should create serializable elements by default

Caveats : limitations being addressed

[430724] Sirius might Serialize a bit too much

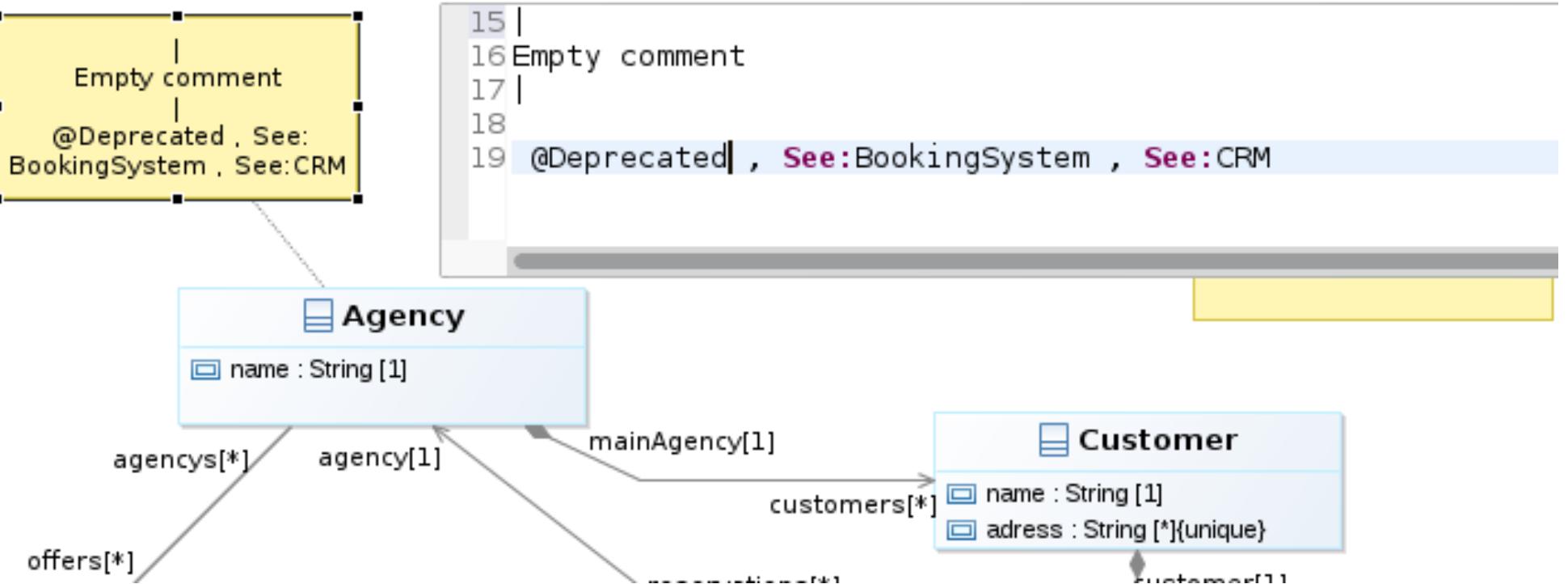
~~[432931] Xtext prevent Serialization with false positives~~

[..] DSL's with Xbase

Embedding



Embedding syntax in a diagram



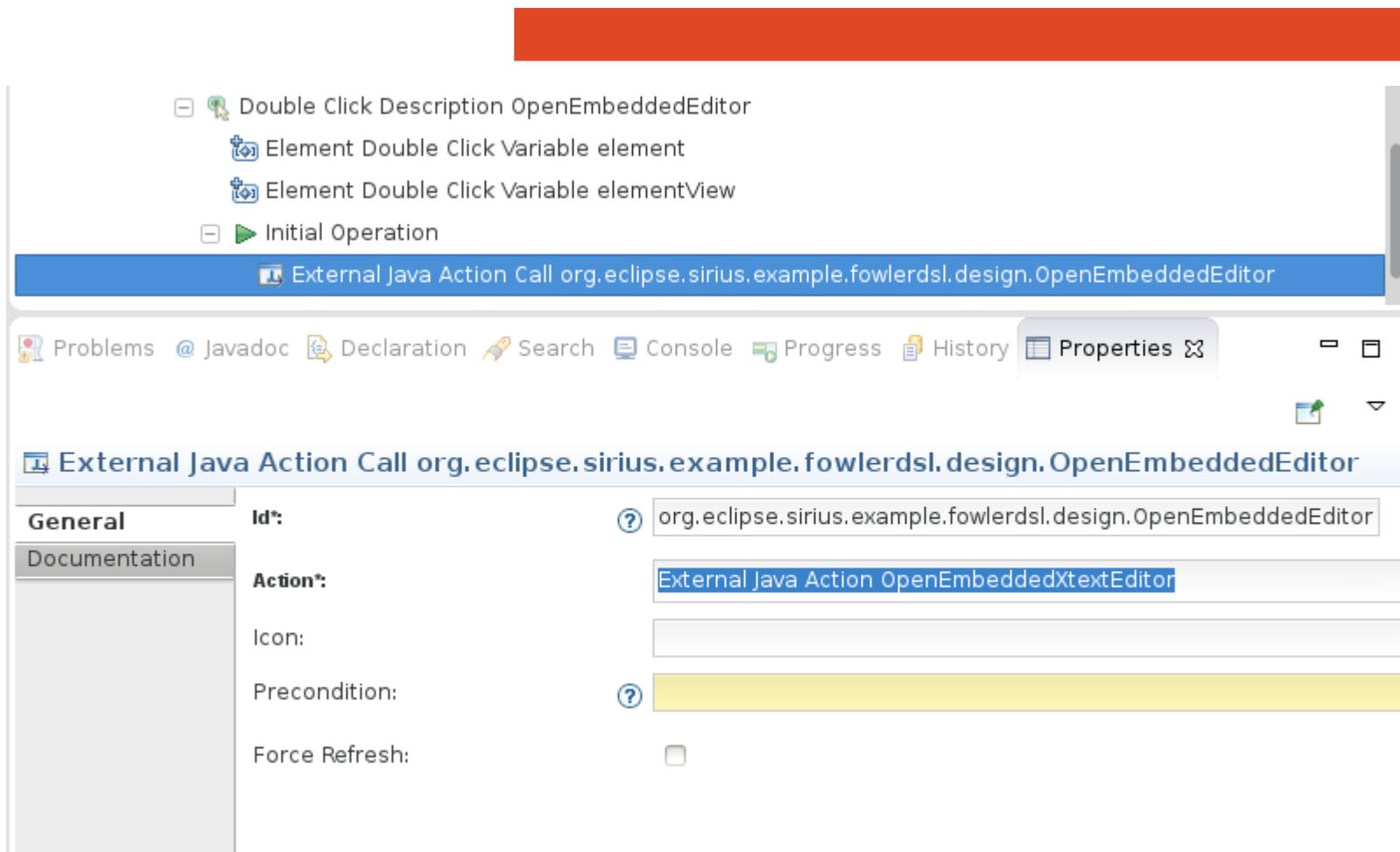
Embedding syntax in a diagram

The diagram illustrates a state machine fragment. On the left, there is a state labeled "newState0" containing an action named "lockPanel". On the right, a code editor window displays the corresponding UML-like syntax:

```
71 state newState0
72     actions {
73         lockPanel |
74     }
75
76 end
```

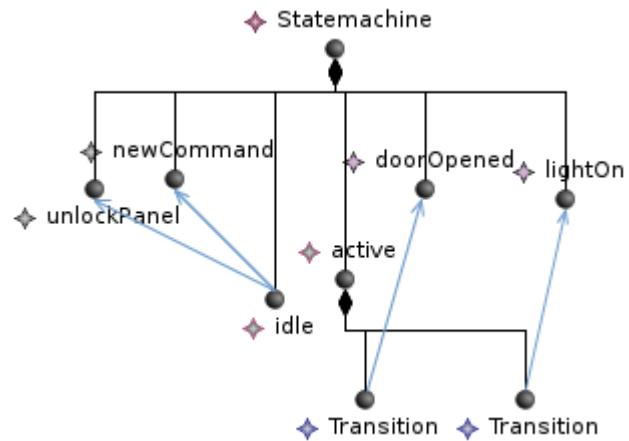
The code is color-coded: "state", "actions", and "end" are in purple, while "lockPanel" is in blue, indicating it is a reference to the action defined in the state.

How To ?



Example and « base framework » here :
<https://github.com/ObeoNetwork/Xtext-viewpoint-integration>

Behind the Scene



```
commands unlockPanel UP lockPanel
end

state idle
actions {
    unlockPanel newCommand
}
doorClosed => active
end

state active
lightOn => waitingForDraw
doorOpened => waitingForDraw
end

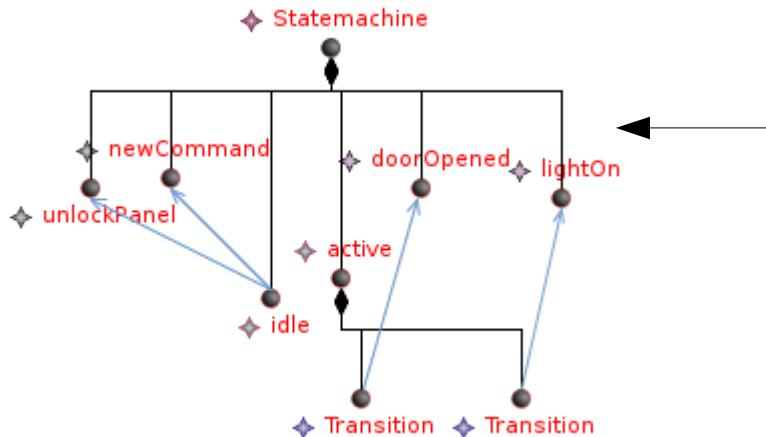
state Nooo
doorClosed => active
lightOn => unlockedPanel
end

state waitingForDraw
actions {
    unlockPanel
}
doorOpened => newStateA
end

state unlockedPanel
end
```

```
29 state waitingForDraw
30     actions {
31         unlockPanel
32     }
33
34     doorOpened => newStateA
35 end
```

Behind the Scene



```
1 commands unlockPanel UP lockPanel
2 end

3 @state idle
4   actions {
5     unlockPanel newCommand
6   }
7   doorClosed => active
8 end

9 @state active
10  lightOn => waitingForDraw
11  doorOpened => waitingForDraw
12 end

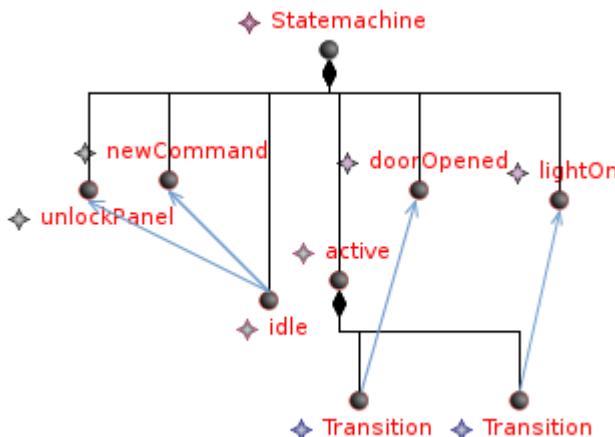
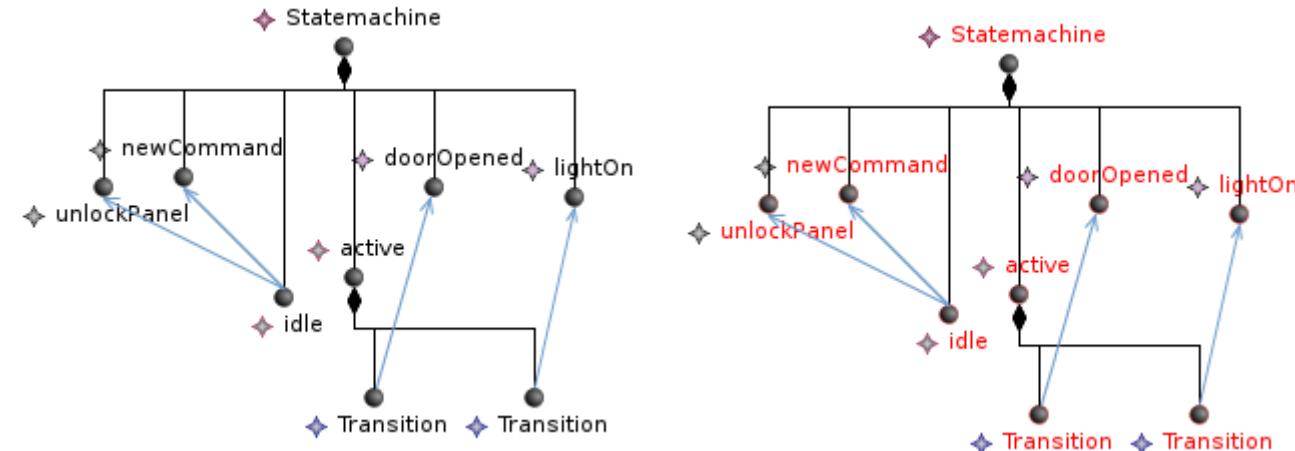
13 |
14 @state Nooo
15  doorClosed => active
16  lightOn => unlockedPanel
17 end

18 @state waitingForDraw
19   actions {
20     unlockPanel
21   }
22   doorOpened => newStateA
23 end

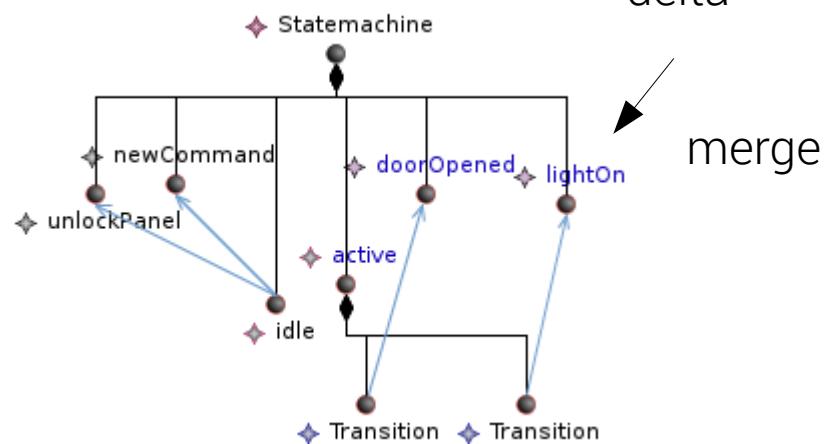
24 @state unlockedPanel
25 end
```

```
29 state waitingForDraw
30   actions {
31     unlockPanel
32   }
33
34   doorOpened => newStateA
35 end
```

Behind the Scene



delta



merge

Caveats : Merging

```
private void reconcile(Resource resourceInSirius,
                      XtextResource resourceInEmbeddedEditor) {
    try {

        IComparisonScope scope = new DefaultComparisonScope(
            resourceInSirius, resourceInEmbeddedEditor, null);
        final Comparison comparison = EMFCompare.builder().build()
            .compare(scope);

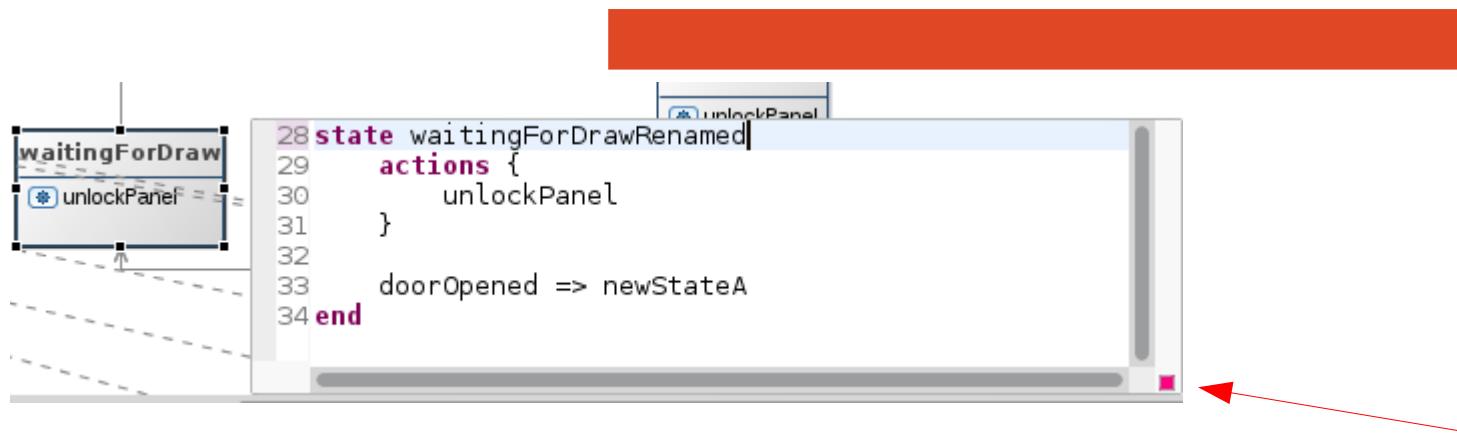
        IMerger.Registry mergerRegistry = EMFCompareRCPPPlugin.getDefault()
            .getMergerRegistry();
        final IBatchMerger merger = new BatchMerger(mergerRegistry);

        final TransactionalEditingDomain editingDomain = TransactionUtil
            .getEditingDomain(originalResource);
        editingDomain.getCommandStack().execute(
            new RecordingCommand(editingDomain,
                "update resource after direct text edit") {

                @Override
                protected void doExecute() {
                    merger.copyAllRightToLeft(
                        comparison.getDifferences(),
                        new BasicMonitor());
                }
            });
    } catch (Exception e) {
        Activator.LogError(e);
    }
}
```

Only as good as how the content matching perform
Bad matching : nodes are re-created in diagram

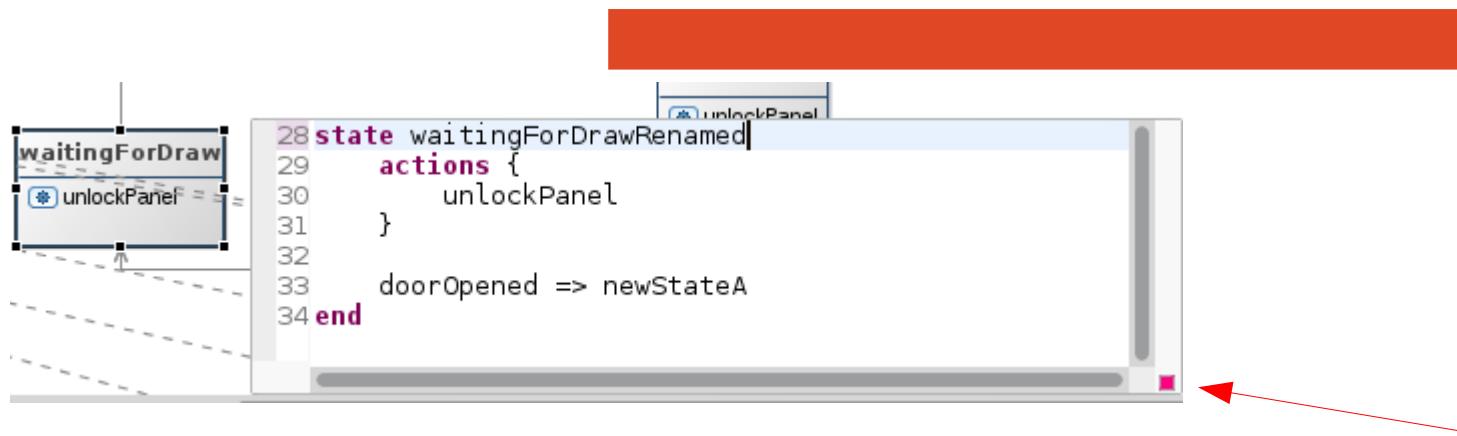
Caveats : Cross-References Consistency



Model is parseable but **not resolvable**

Either prevent it* or warn the user !

Caveats : Cross-References Consistency



Model is parseable but **not resolvable**

Either prevent it* or warn the user !

These caveats should be considered **upfront**, it either impact the user experience or your DSL itself

Takeaways



- Sirius is there and works with Xtext
- If you want to embed : think about what you expect
- Get involved : give feedback and or patches on the embedding layer

<https://github.com/ObeoNetwork/Xtext-Sirius-integration>

What Next ?



Eclipse Luna

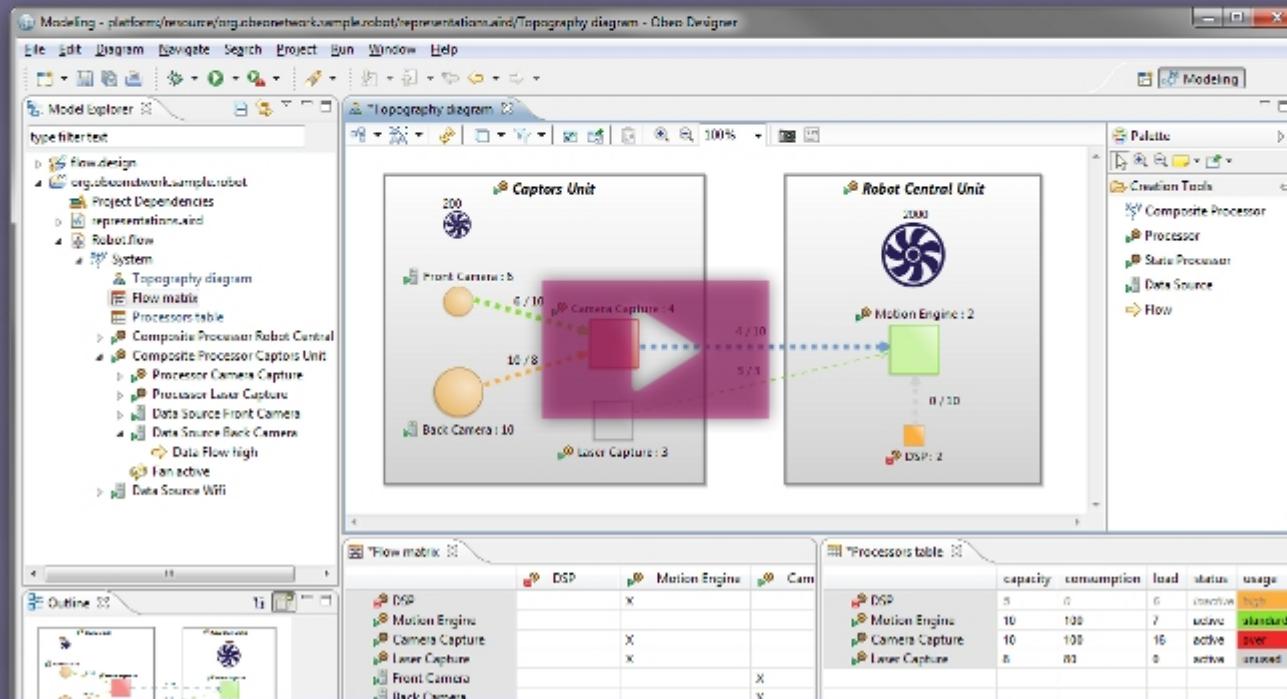
- Sirius goes 1.0
- Compartments (preview)
- API and Ecore model cleanup
- EcoreTools 2, UML Designer 4 are based on Sirius

Sirius 2.0 (*) :

- Runtime lazyness
- Compartments

() Model migrations are taken care of by Sirius*

The easiest way to get your own modeling tool

[GET SIRIUS NOW](#)[DOWNLOAD](#)