• Focused on **domain specific modeling** technologies

• 50+ people working on products and customer projects

• Services: training, consulting and Open Innovation

• Tool vendor

www.oobeo.fr
«Xtext brings the capability of defining fairly quickly a **textual** editor **supporting your domain specific concepts**.»

«Sirius brings to anybody the capability of defining fairly quickly **graphical**, tree or table editors **supporting your domain specific concepts**.»
Textual + Graphical?

Text is better:
- at operations and sequences
- when combining (expressions)
- ...
- for some users

Diagram is better:
- relationships
- analyzing a design
- ...
- for some users

Two complementary approaches
The world of **Sirius**
The world of **Sirius**
The world of **Sirius**
The world of Sirius
The world of **Sirius**

```
Statemachine A
  Event E
    State A1
    State A2

Statemachine B
  State B1
```

- a.xmi
- b.xmi
- bfrag.xmi
- representations.xmi
- save/load

**workspace**
The world of **Xtext**

```plaintext
myModel.statemachine

 salvation/load

workspace
```
The world of **Xtext**

```
myModel.statemachien
```
The world of Xtext
The world of **Xtext**

```
events
doorClosed drawOpened
lightOn reset
doorOpened panelClosed
newEvent CD
end

{} commands
unlockPanel UP
lockPanel LP
lockDoor LD
unlockDoor UD
newCommand NC
end

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

myModel.statemachine
```

save/load

workspace

scope/indexer
Unexpected situations?

Not so frequent situations for Sirius:
- thousands of resources in a single project
- instance changes its URI/loose identity
- serialization is not possible

Not so frequent situations for Xtext:
- state is not serializable
- model graph could not have been constructed through Xtext
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 2.7.2**
** Sirius 3.0.0 (pre M6)**
**Eclipse Luna SR2**

https://github.com/ObeoNetwork/Xtext-Sirius-integration
Caveats: parsing errors
Caveats: parsing errors
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

Tools cleaning up the graph (Association like instances..)
Instantiate an XtextResourceSet
Configure the ResourceSet with the project.
Change the saving policy
Caveats: **limitations addressed**

- [430724] Sirius might serialize a bit too much
- [432931] Xtext prevent serialization with false positives
- [448304] Xtext is not updating cross-ref names
- [458841] Xcore model can't resolve references to standard types.
The Best of both
Embedding syntax in a diagram

```java
final Comment comment = (Comment)((IDiagramElementEditPart)graphicalEditPart)
    .resolveTargetSemanticElement();

XtextAttributeEmbeddedEditor embeddedEditor = new XtextAttributeEmbeddedEditor(
    graphicalEditPart, getInjector()) {

    @Override
    protected void updateModel(XtextResource parseResult, String editedText) {
        comment.setBody(editedText);
    }
}```
Example and « base framework » here:
https://github.com/ObeoNetwork/Xtext-sirius-integration
Behind the Scene

```plaintext
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
```
Behind the Scene

```
commands unlockPanel UP lockPanel
end

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

state active
    lightOn => waitForDraw
    doorOpened => waitForDraw
end

state Nboo
    doorClosed => active
    lightOn => unlockedPanel
end

state waitForDraw
    actions {
        unlockPanel
    }
    doorOpened => newStateA
end

state unlockedPanel
end
```
Behind the Scene
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.
Caveats: **Merging**

```java
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
React to **external changes**

Sirius detects changes in the workspace:

Domain model is reloaded when externally changed.

**Experimental** approach:
- Do not reload
- Compute the delta between the resource on the disk and the loaded state and merge differences.
How to?

Example and «experimental framework» here: https://github.com/mporhel/Xtext-sirius-integration

Core modification in Sirius not yet integrated but available here: https://git.eclipse.org/r/#/c/43489/
Caveats: **Merging**

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

« Validate action » in your Xtext Editor

- Marker in the Problems view
- Referencing the file
- Double click opens your Xtext Editor

« Validate diagram » action in Sirius diagram editors

- Call semantic validation rules and rules defined in the VSM
- Marker in the Problem view
- Referencing the representations files, and the validated representation
- Double click opens your Sirius editor
External validation **markers**

```
sample.statemachine

11 LOCKPanel LP
12 lockDoor LD
13 unlockDoor UD
14 end

18 state idle
19  actions {unlockDoor lockPanel}
20     doorClosed => Active
21 end

27 state Active
28     drawOpened => waitingForLight
29     lightOn  => waitingForDraw
30 end

36 state waitingForLight
37     lightOn  => unlockedPanel
38 end

32 state waitingForDraw
33     drawOpened => unlockedPanel
34 end

38 state unlockedPanel
39     actions {unlockPanel lockDoor}
40     panelClosed => idle
41 end
```
How to?

```java
org.eclipse.gmf.runtime.diagram.ui.decoratorProviders
  FowlerValidationDecoratorProvider (decoratorProvider)

"org.eclipse.xtext.example.fowlerdsl.ui.statemachine.check.fast";
"org.eclipse.xtext.example.fowlerdsl.ui.statemachine.check.normal";
"org.eclipse.xtext.example.fowlerdsl.ui.statemachine.check.expensive";

for (String markerType : getMarkerTypes()) {
    IMarker[] markers = null;
    try {
        markers = resource.findMarkers(markerType, true, IResource.DEPTH_INFINITE);
    }
    finally {
        if (markers != null) {
            for (IMarker marker : markers) {
                String attribute = marker.getAttribute("URI_KEY", ""); //NON-NLS-1$
                if (attribute.equals(uri.toString())) {
                    int nextSeverity = marker.getAttribute(IMarker.SEVERITY, IMarker.SEVERITY_INFO);
                    Image nextImage = getImage(nextSeverity);
                    if (foundMarker == null) {
                        foundMarker = marker;
                        toolTip = new Label(marker.getAttribute(IMarker.MESSAGE, ""), //NON-NLS-1$
                                             nextImage);
                    } else {
                        toolTip = null;
                    }
                }
            }
        }
    }
}
```
Takeaways

- Sirius tuning: refresh and saving policy
- Think about the referencing
- Add editor feedback when not serializable
- Design create/update/delete operations to stay serializable
- When embedding, display elements which should not have cross.refs (or take care of those)
- EMF Compare can help
Takeaways

- As long as it's an EMF model Sirius will handle it.
- If you want to embed: think about what you expect.
- Get involved: give feedback and or patches on the embedding, reload or validation layers.

https://github.com/ObeoNetwork/Xtext-Sirius-integration
What **Next ?**

Sirius **1.0.2** released with **Luna SR2**

**Sirius 2.0 (**)**

- UX improvements for diagrams and layouts
- Performance, scalability
- Ecore Tools 2, UML Designer 4 are based on Sirius
- 2.0.4 released on Feb 12, 2015 (~40 Bugzillas delivered since 2.0.0 in October)

**Eclipse Mars :**

- Sirius goes **3.0 (**)**
- Performance, Scalability: hitting the 1 Million mark
- Reduced memory footprint
- Edges connected to real outline of shapes
  (images with transparency, basic shape style)

(*) Model migrations are taken care of by Sirius
Other Talks

• Tuesday
  • Arduino designer: the making of! – Mélanie
    • Grand Peninsula EFG - 13:30 to 14:05

• Wednesday
  • Sirius + Xtext = ♥ - Maxime
    • Grand Peninsula D - 13:30 to 14:05
  • Poster session – Etienne, Maxime, Mélanie
    • Atrium - 18:00 to 19:30

• Thursday
  • On the road to Mars with Mars, thanks to Polarsys – Thales
    • Harbour AB - 15:15 to 15:50
Sirius  http://www.eclipse.org/sirius/
Evaluate the sessions

Sign in: www.eclipsecon.org