

Integrating Xtext and Sirius: Strategies and Pitfalls

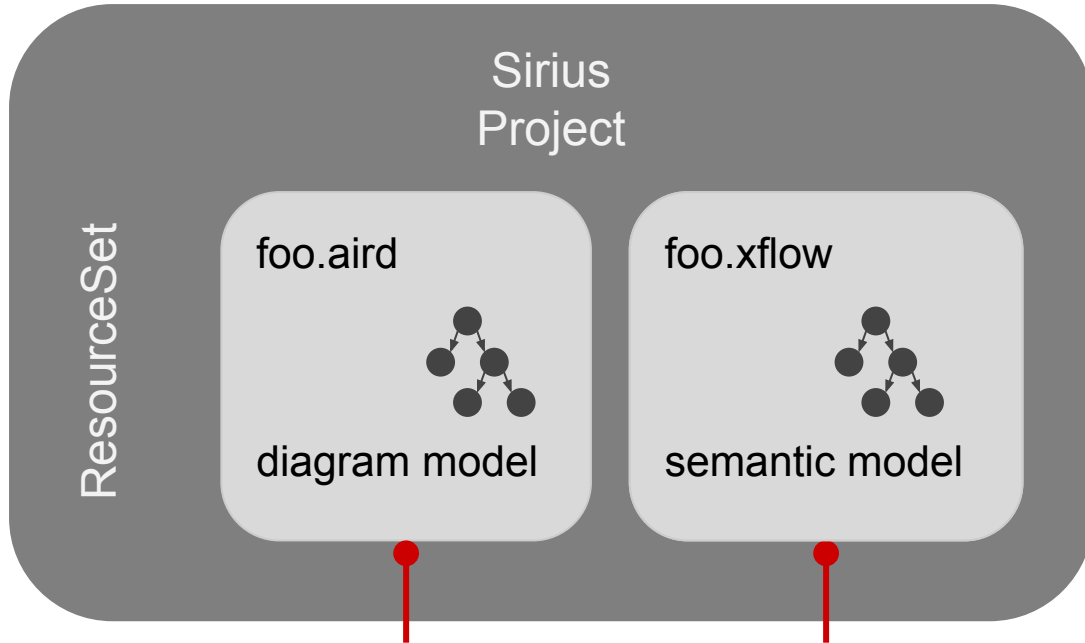
Cédric Brun (Obeo) – Jan Köhnlein (TypeFox)

Agenda

- Three modes of integration
 - Persisting the model using Xtext
 - References between Xtext and Sirius models
 - Embedding an Xtext editor in Sirius
- Conclusion

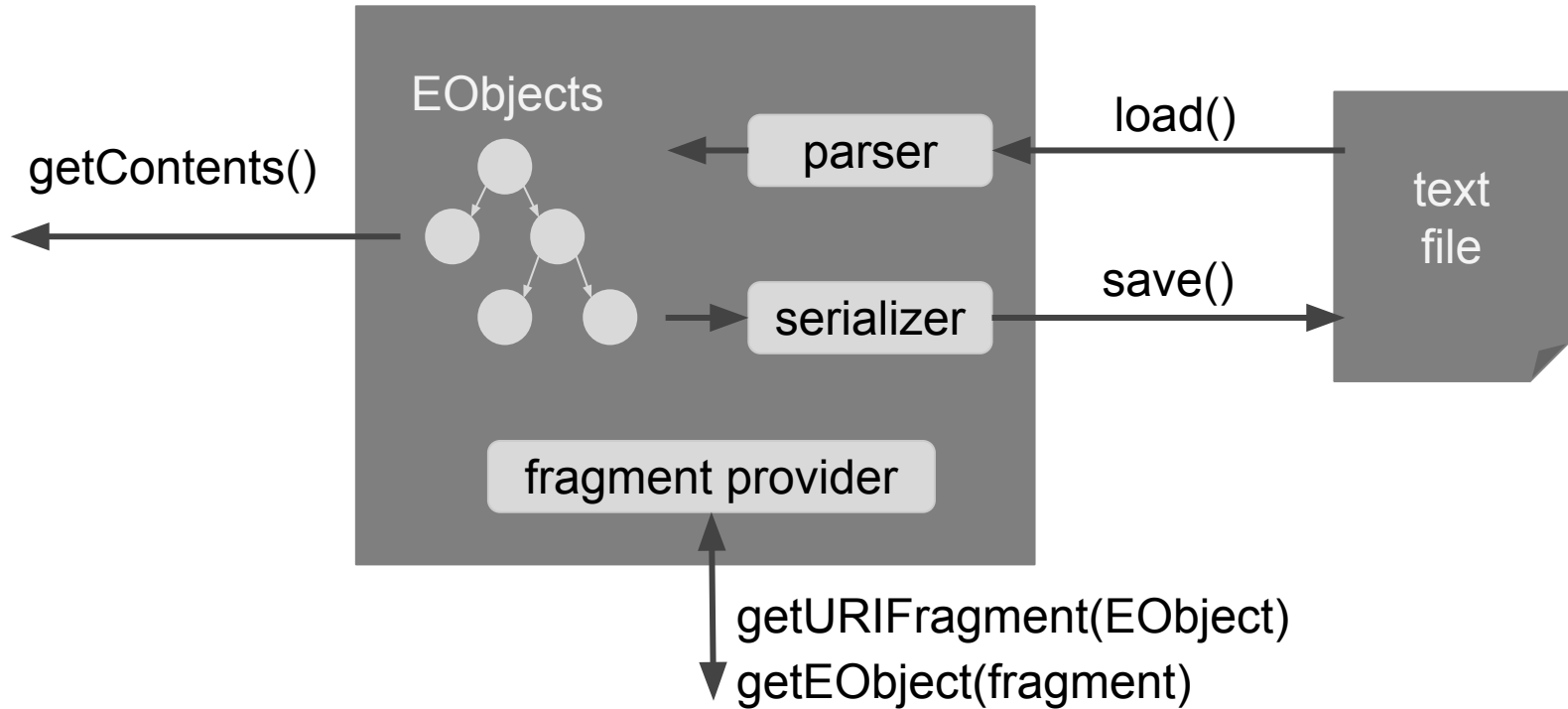
Persisting the Model Using Xtext

Diagram Model vs Semantic Model



by default persisted as XMIResources

XtextResource



Persist the Semantic Model with Xtext

Sync through filesystem

The screenshot displays an IDE with two main windows. The left window, titled 'teststatemachine', contains Xtext code for a state machine. The right window, titled 'new State Machine Diagram', shows a UML State Machine Diagram that visually represents the code. Below these windows is a 'Properties' panel for the 'State waitingForDraw'.

```
events doorClosed CLOSED drawOpened DOPENED lightOn LON reset RST doorOpened DOORO panelC
end
commands unlockPanel UP lockPanel LP lockDoor LD unlockDoor UD
end
state idle
actions {
  lockDoor unlockPanel
}
doorClosed => active
end
state active
drawOpened => waitingForLight
lightOn => waitingForDraw
end
state waitingForLight
actions {
  unlockDoor
}
lightOn => unlockedPanel
end
state waitingForDraw
actions {
  unlockPanel
}
drawOpened => unlockedPanel
end
state unlockedPanel
panelClosed => idle
end
```

The State Machine Diagram shows the following states and transitions:

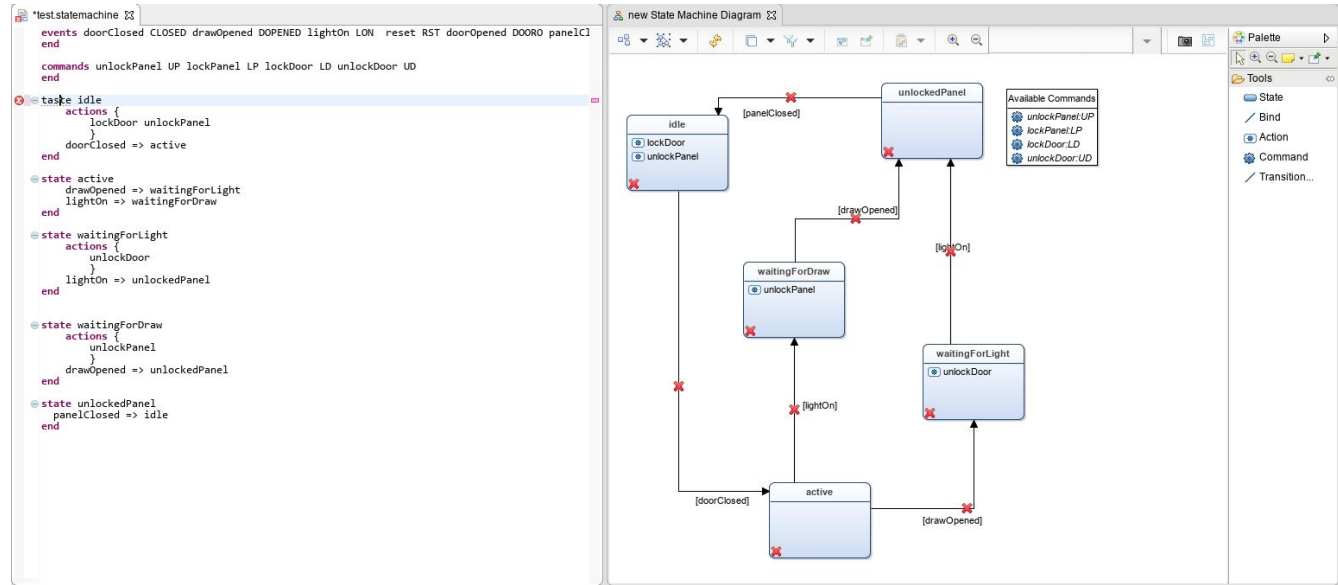
- idle** (State): Contains actions 'lockDoor' and 'unlockPanel'. Transitions: '[panelClosed]' to 'unlockedPanel', '[doorClosed]' to 'active'.
- active** (State): Transitions: '[drawOpened]' to 'waitingForDraw', '[drawOpened]' to 'unlockedPanel', '[lightOn]' to 'waitingForLight'.
- waitingForDraw** (State): Contains action 'unlockPanel'. Transitions: '[drawOpened]' to 'unlockedPanel', '[lightOn]' to 'waitingForLight'.
- waitingForLight** (State): Contains action 'unlockDoor'. Transitions: '[lightOn]' to 'unlockedPanel'.
- unlockedPanel** (State): Contains no actions. Transitions: '[panelClosed]' to 'idle', '[lightOn]' to 'waitingForLight'.

The 'Properties' panel for 'State waitingForDraw' shows the Xtext code for this state:

```
state waitingForDraw
actions {
  unlockPanel
}
drawOpened => unlockedPanel
end
```

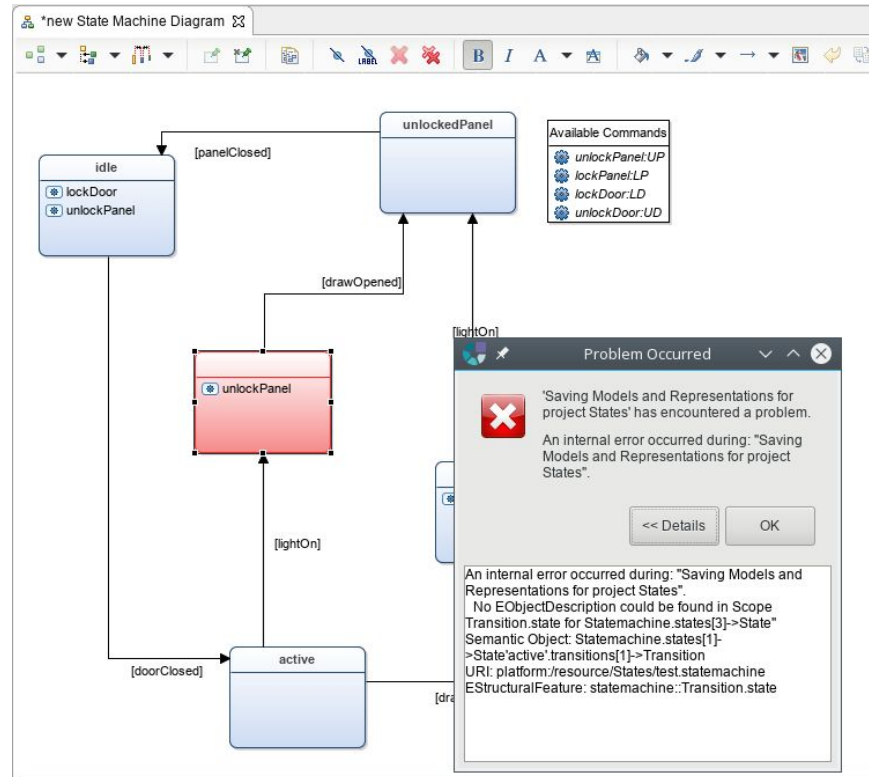
Diagram Refresh Strategy

- Automatic Refresh (default)
 - lead to diagram re-init when the model is un-parseable
 - diagram content gets lost
- Setup a specific strategy to trigger the refresh



Grammar Adaptations

- Avoid errors when the user tries to save
- Not every string is an ID
 - Consider using STRING rule instead of ID in the grammar
- Lose grammar, strict validation
- Tools should create serializable elements by default
- Highlight offending elements in diagram



Concurrent Dirty States

Prevent editing in the other syntax when one is dirty

The image shows a state machine editor interface with two main panes. The left pane displays the state machine code for 'teststatemachine', and the right pane displays a corresponding state machine diagram. A dialog box is overlaid on the diagram, asking to reload the resource.

```
teststatemachine
events doorClosed CLOSED drawOpened DOPENED lightOn LON reset RST doorOpened DO
end
commands unlockPanel UP lockPanel LP lockDoor LD unlockDoor UD
end
state idle
actions {
  lockDoor unlockPanel
}
doorClosed => active
end
state active
drawOpened => waitingForLight
lightOn => waitingForDraw
end
state waitingForLight
actions {
  unlockDoor
}
lightOn => unlockedPanel
end
state waitingForDraw
actions {
  unlockPanel
}
drawOpened => unlockedPanel
end
state unlockedPanel
panelClosed => idle
end
state additionFromTextEditor
end
```

The state machine diagram on the right shows the following states and transitions:

- idle** (initial state) transitions to **waitingForLight** on `lockDoor` and to **unlockedPanel** on `unlockPanel`.
- waitingForLight** transitions to **unlockedPanel** on `lightOn`.
- waitingForDraw** transitions to **unlockedPanel** on `drawOpened`.
- unlockedPanel** transitions to **idle** on `panelClosed`.
- renamedInDiagram** (a state added from the text editor) transitions to **waitingForLight** on `lightOn` and to **waitingForDraw** on `drawOpened`.

A dialog box titled "Reload the resource?" is displayed over the diagram. The message reads: "The platform:/resource/States/teststatemachine resource has been externally changed, should we reload it?". The dialog has "No" and "Yes" buttons.

An "Available Commands" panel on the right lists the following commands:

- unlockPanel:UP
- lockPanel:LP
- lockDoor:LD
- unlockDoor:UD

References Between Xtext and Sirius Models

Xtext and Sirius Models Combined

Xtext and Sirius are aware of each others models

Cross-references
between them

The screenshot shows a software application with two main panes. The left pane, titled 'exploitation description', displays a graphical model. It features a central box labeled 'in [cultures activities]' containing icons for 'Tractor' and 'People'. Below this, there are icons for 'Massey Ferguson 1', 'John', and 'Old MacDonald'. A 'dedicated to' relationship is shown between the tractor/people box and a 'Crop' box. The 'Crop' box contains icons for 'corn', 'sorgho', and 'wheat'. Below the 'Crop' box, there are three colored boxes representing field areas: a blue box for '130ha : 4 fields', a green box for '100ha : 10 fields', and an orange box for '80ha : 5 fields'. A 'Daily' box with a cow icon is also present, connected to the field boxes. A 'surfaces ratios' chart is visible on the left side of the model.

The right pane, titled 'cultures.activities', displays Xtext code. The code defines activities for 'wheat' and 'sorgho' crops, including 'PLOWH', 'SOW', 'FERTILIZATION', and 'HARVEST'. The code is as follows:

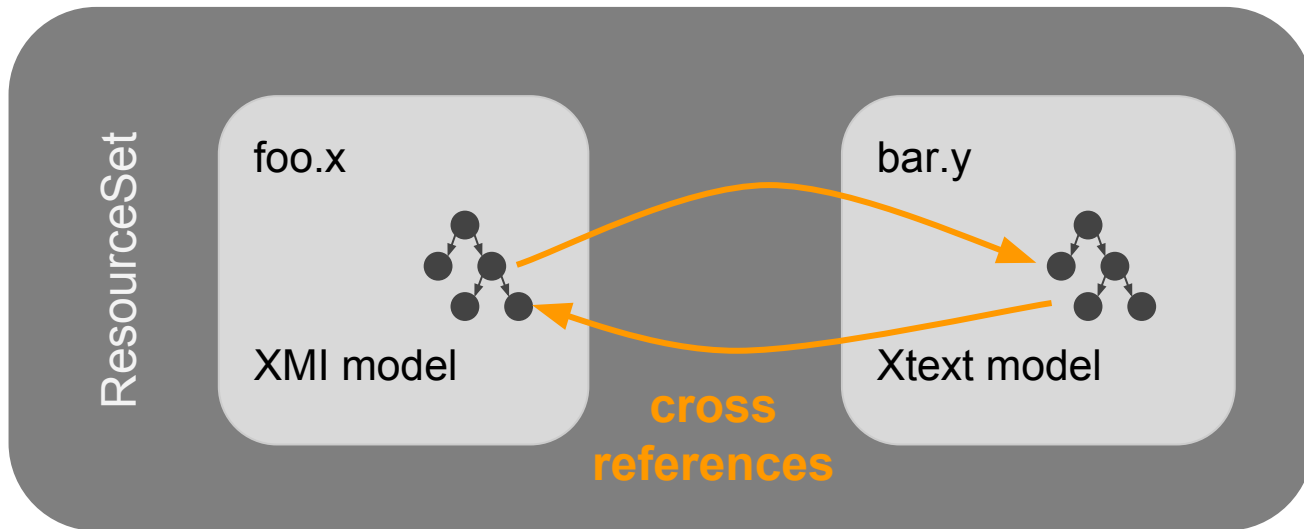
```
] using 1 Tractor and 1 People
activity FERTILIZATION from 15 mar to 15 jun [
  after SOW is done since 30 days &&
  no rain since 1 days
] using 1 Tractor and 1 People
activity HARVEST from 1 sept to 30 sept [
  grain is "mature" &&
  after SOW
] using 1 Tractor and 2 People
}

culture wheat {
  activity PLOWH from 1 sept to 30 sept [
    no rain since 3 days
  ] using 1 Tractor and 1 People
  activity SOW from 1 oct to 31 oct [
    after PLOWH &&
    no rain since 3 days &&
    temperature > 5°C
  ] using 1 Tractor and 1 People
  activity FERTILIZATION from 1 feb to 28 feb [
    after SOW is done since 30 days &&
    no rain since 1 days
  ] using 1 Tractor and 1 People
  activity HARVEST from 1 jun to 30 jun [
    grain is "mature" &&
    after SOW
  ] using 1 Tractor and 1 People
}

culture sorgho {
  activity PLOWH from 1 jan to 30 mar [
    no rain since 3 days
  ] using 1 Tractor and 1 People
  activity SEMIS from 1 may to 15 may [
    after PLOWH &&
    no rain since 5 days &&
    temperature > 5 °C
  ] using 1 Tractor and 1 People
  activity HARVEST from 1 oct to 15 oct [
    grain is "mature"
  ] using 1 Tractor and 1 People
}

resource Tractor
resource People
```

EMF Cross-references



XMI uses URIs: href=" <resource URI>#<fragment> "
e.g. href=" bar.y#ab378f1789d1 "

Names vs URIs

I am giving this talk with Cédric.

Name-based linking (Xtext)

I am giving this talk with ECE2017#116.

URI-based linking (Sirius)

Differences

Resolution mechanism

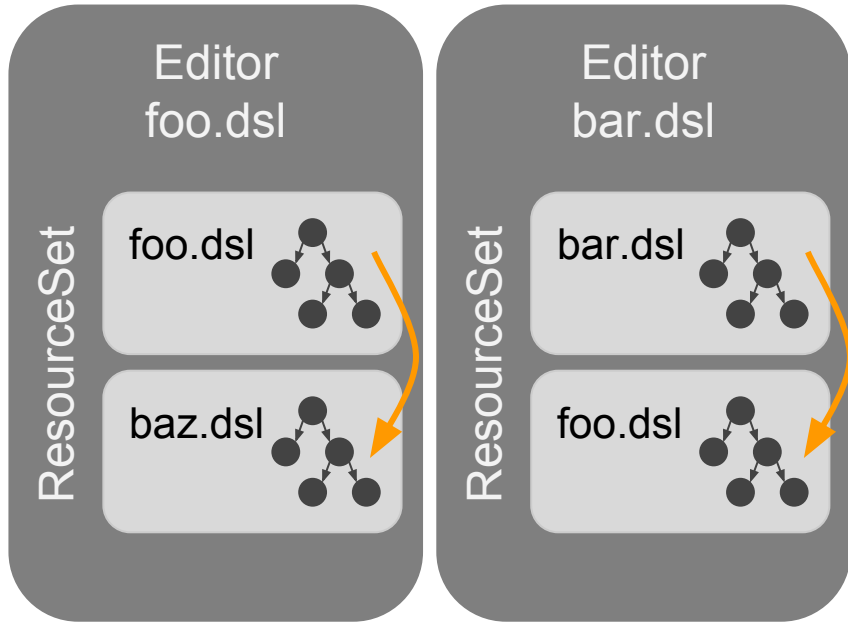
Coupling of resources

Syntax errors

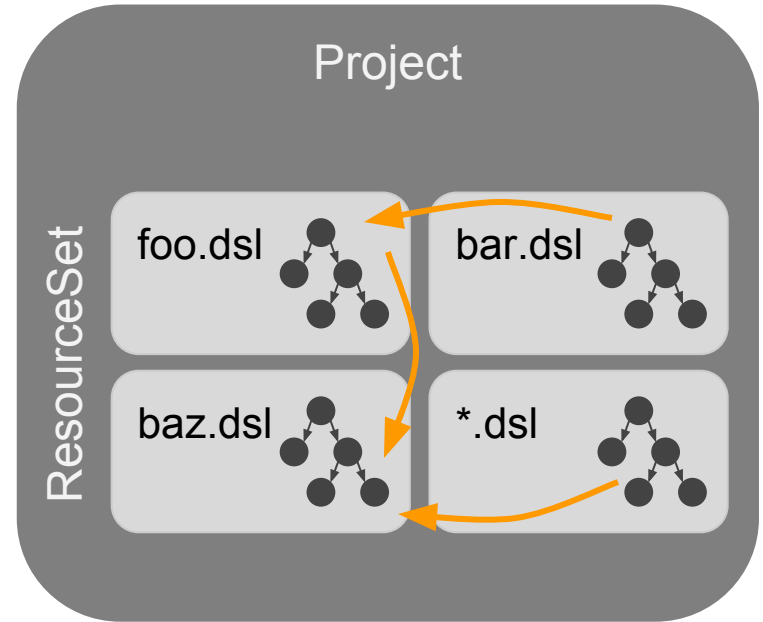
Rename

Suitability for users

Models And Identities



Xtext



Sirius

Make Sirius Refer to Xtext

- Implement an IFragmentProvider
 - make name (and type) part of the XMI-ID
- Implement Xtext support for *.aird files
 - allows Sirius to participate in Xtext rename refactoring
- Navigate to Xtext editor from the diagram
- Define fixed synchronization points
- Compare based on compare & merge instead of plain reload

Make Xtext Refer to Sirius/XMI

- Implement Xtext support for the XMI model
 - make elements referable from Xtext
 - make cross-links openable
 - participate in Xtext refactorings
- Navigate to Sirius editor from Xtext
- Rename on the Sirius side must trigger a rename refactoring in Xtext

Either way

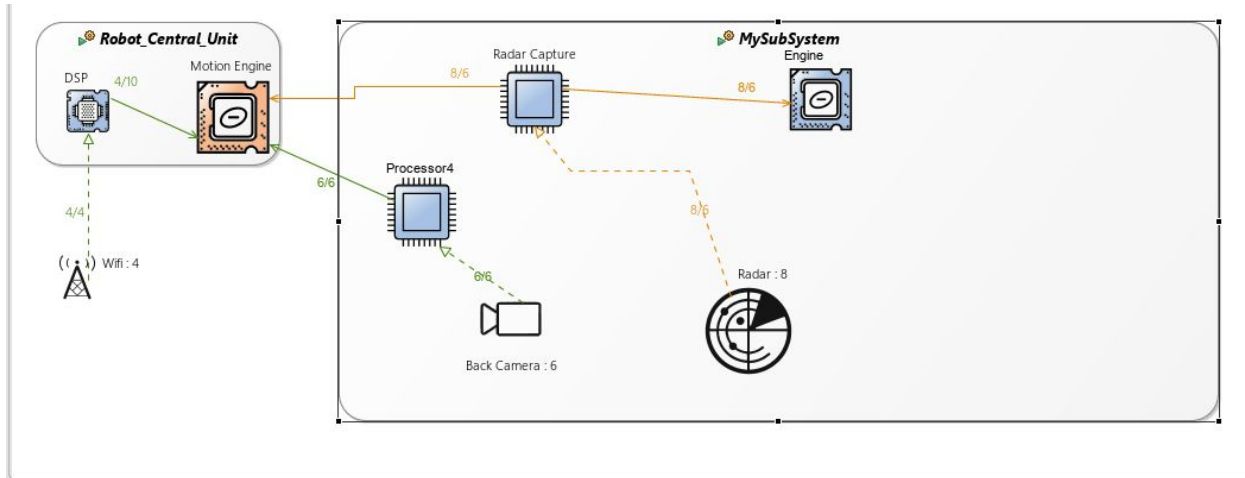
- Cross-referencing will rely on the Xtext index
 - Xtext nature on project + build-automatically must be on
 - check the index with CTRL-SHIFT-F3
- Dirty-state integration is tricky
 - Refs from Xtext are not updated correctly when Sirius model changes
 - Avoid Xtext rename refactoring when diagram has unsaved changes

Embedding an Xtext Editor into Sirius

Embedding Xtext

XtextEmbeddedEditor +
Custom properties widget

Store Xtext model in a
string-valued attribute



Properties <> Interpreter Error Log

Composite Processor MySubSystem

Main

Routing

Semantic

Style

Appearance

Rules

```
Case MySubSystem.temperature >= 60 °C :  
^ crit DSP Processor DSP  
  Engine  
  MySubSystem  
  Processor4  
  Radar  
  Robot_Central_Unit  
  Wifi  
  1 - Value
```

Press 'F2' for focus

Requires Pretty Deep Integration

- Configuring XtextEmbeddedEditor widget
- Syncing Transactions
- Using Sirius dirty state instead of the last-saved state
- Rename refactoring will be hard

Conclusion

You can integrate Xtext and Sirius

- Use EMF as the base of any kind of integration
- Basic level of integration is quite easy
- Deep integration can be tricky, keep an eye on
 - Linking cross-references
 - Using resource sets
 - Transactions
 - Consistency
- But in the end you'll get a great tool mixing textual and graphical representations

Examples are online

- FowlerDSL: <https://github.com/ObeoNetwork/Xtext-Sirius-integration/>
- Farming (Gemoc) : <https://github.com/gemoc/farmingmodeling>
- Flow: <https://github.com/ObeoNetwork/Flow-Designer>

Blog post [“Linking Xtext Models With EMF Models”](#)

And a White Paper “**Xtext / Sirius - Integration, The Main Use-Cases**” coming soon.



eclipsecon
Europe 2017



OSGi™
Community Event

2017

Evaluate the Sessions

Sign in and vote at eclipsecon.org

- 1 0 + 1