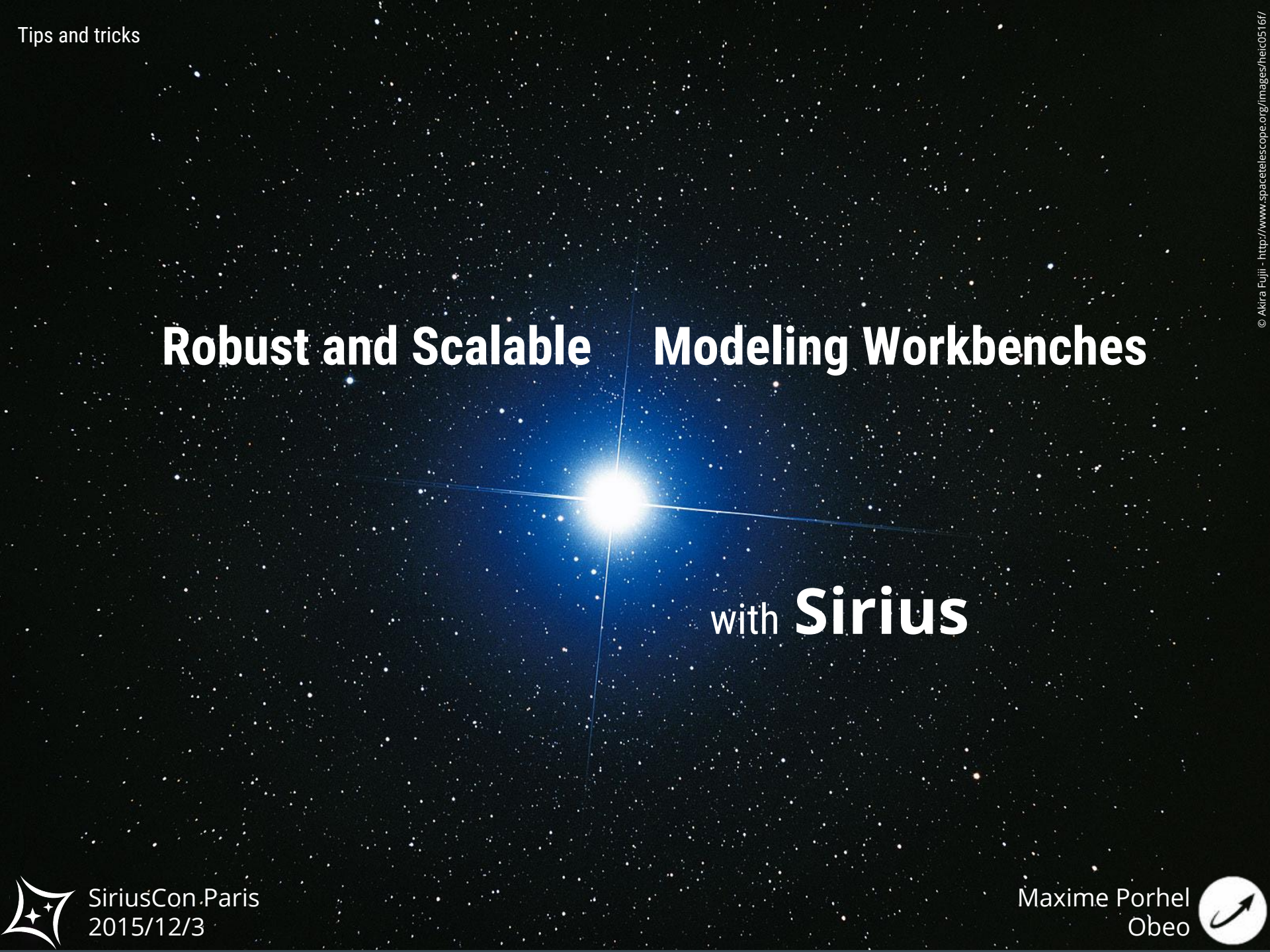
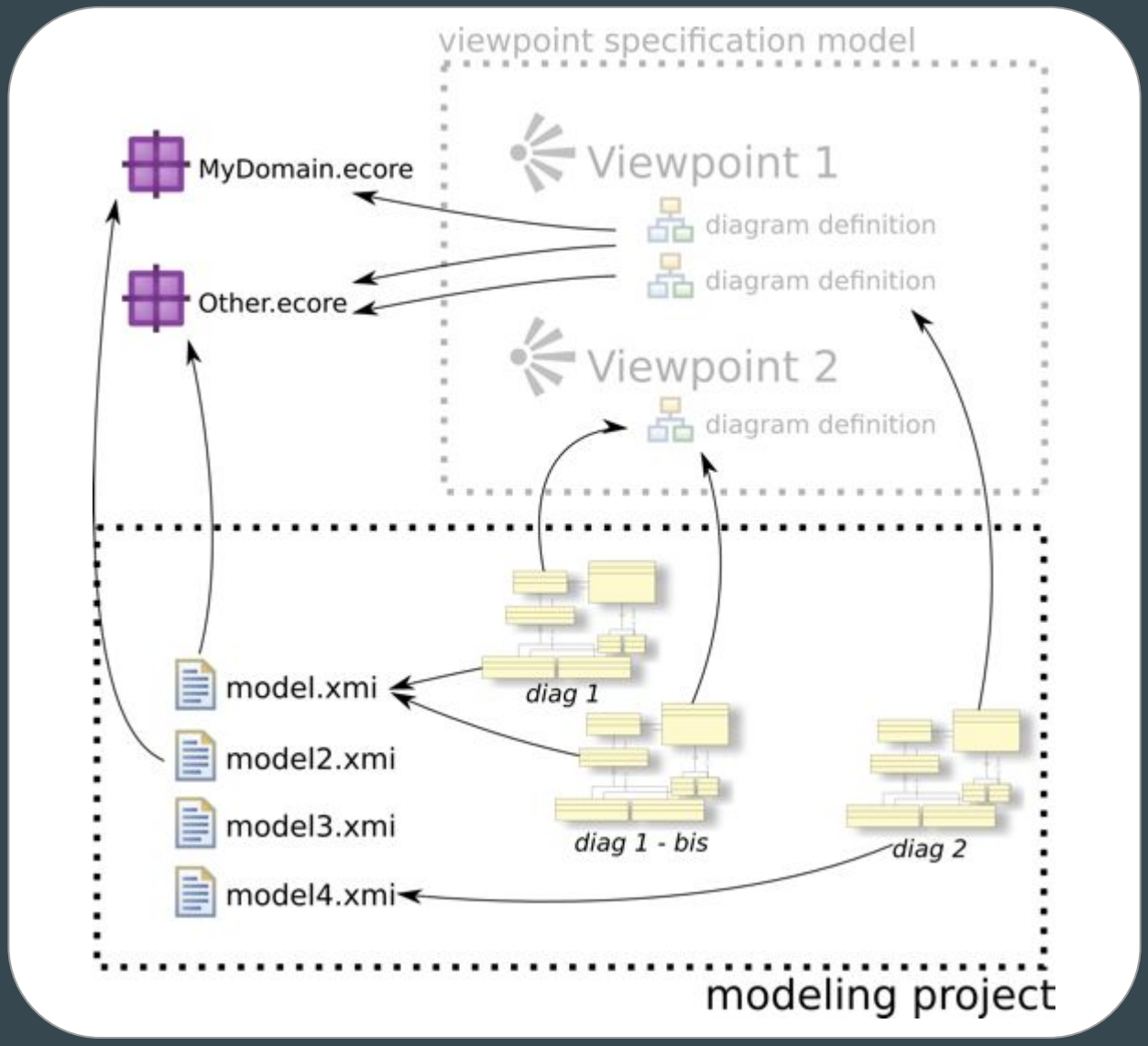
Robust and Scalable Modeling Workbenches with Sirius

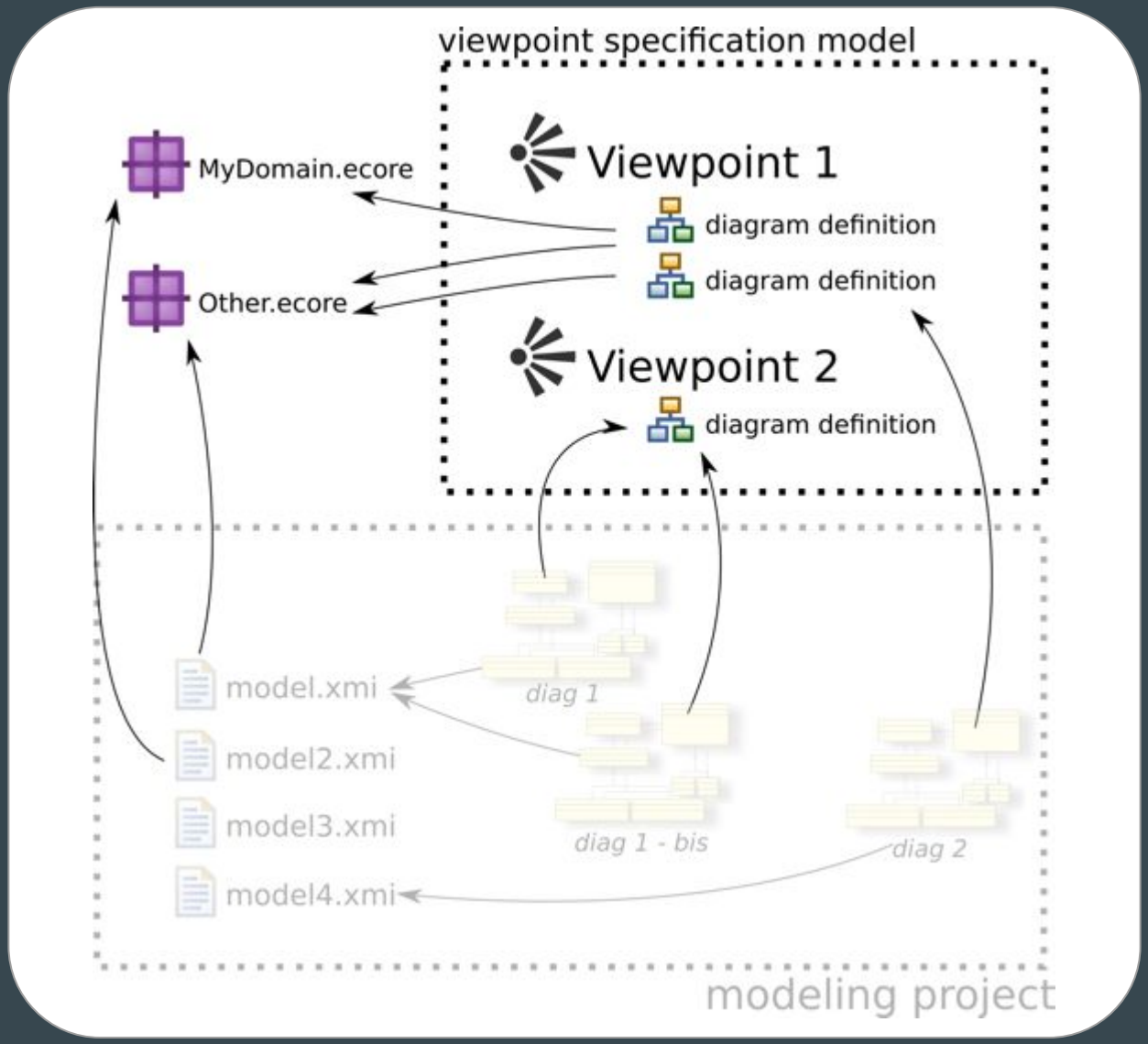


Maxime Porhel
Obeo

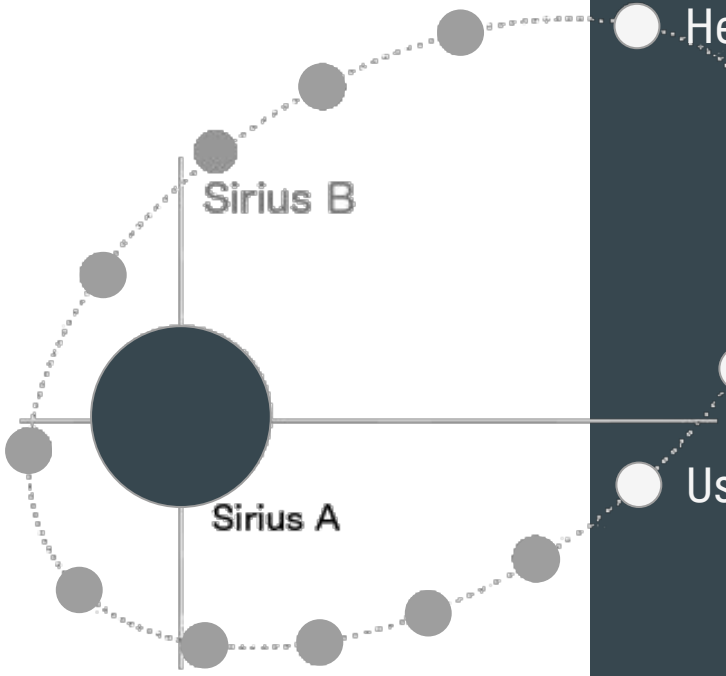
A dedicated tooling



A dedicated tooling

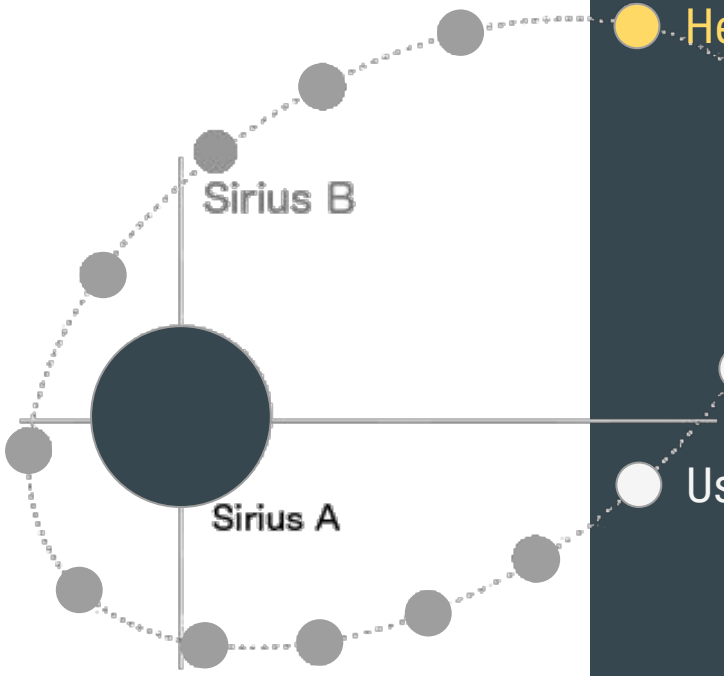


Outline



- Help Sirius find the elements to display
- Synchronization options and advanced tools
- Additional mappings and tools contribution
- Style and color customization
- Use the best query language for the task

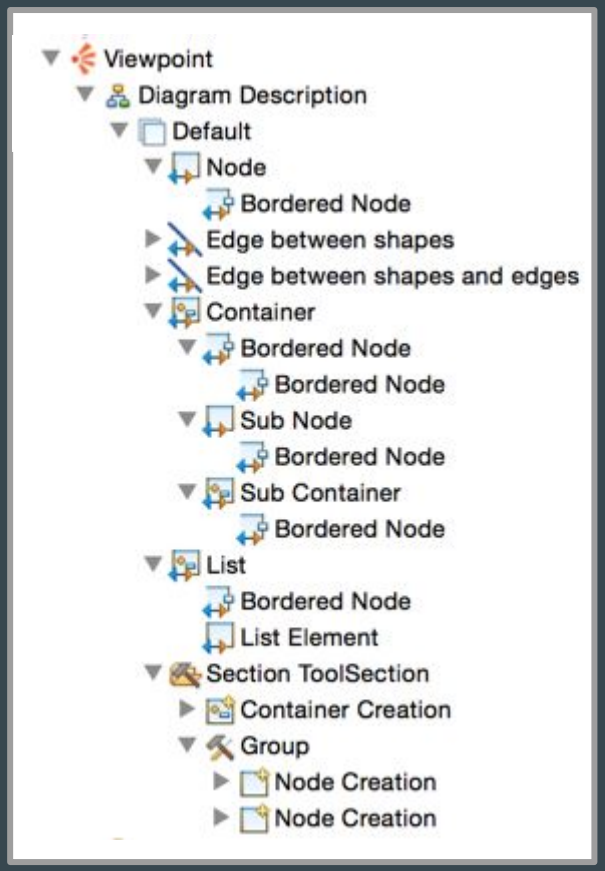
Outline

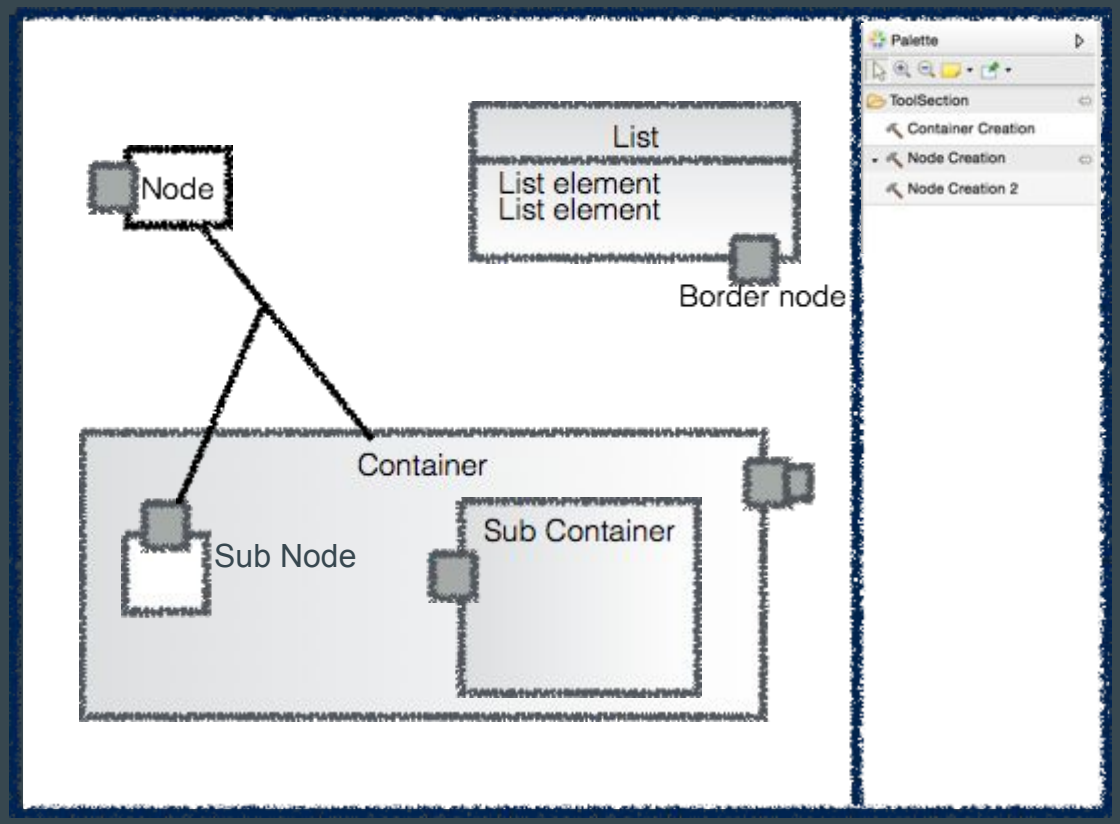


- Help Sirius find the elements to display
- Synchronization options and advanced tools
- Additional mappings and tools contribution
- Style and color customization
- Use the best query language for the task

Viewpoint specification model

Mappings and tool declaration



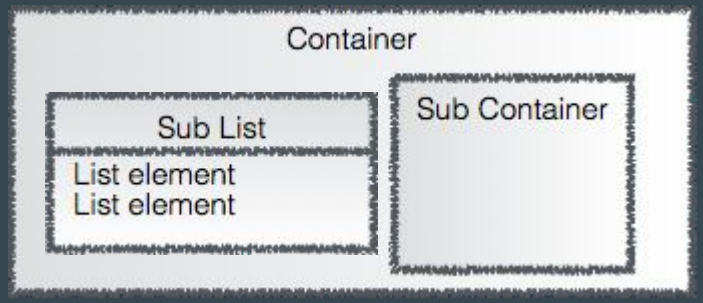


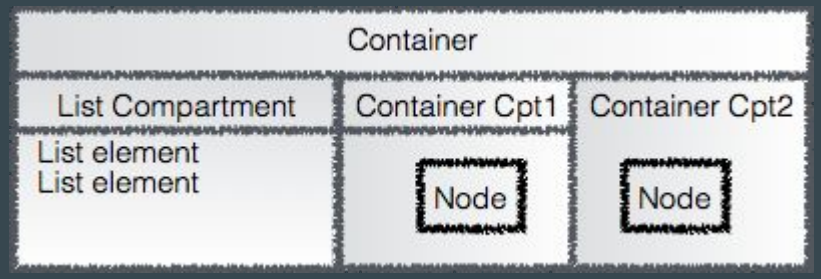
Viewpoint specification model

Several kind of containers

Supported children presentations:

- **Free form**
- **List**
  - node mappings to define list elements
- **Compartments**
  - container mappings to define compartments
  - fixed or dynamic
  - **vertical / horizontal stacks**





Viewpoint specification model

Naive approach

- Domain Class

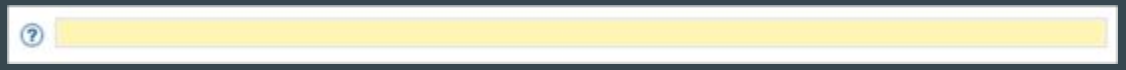
- No Semantic Candidates Expression

- Precondition expression to filter

Note:

- **Green**: EClass qualified name
- **Yellow**: interpreted expression

Interpreted expression



- Tooltip: the expected type of result and the available variables
- Completion on empty expression: available interpreters

- **var:** direct access to Sirius variables
- **feature:** direct access to the named features of the current element
  (and EMF pseudo-features)
- **service:** direct call of a Java method
  (that follows some naming conventions, see documentation)
- **aql:** Acceleo Query Language
  (introduced with Sirius 3.0, **recommended** since 3.1)
- **[/]**: Acceleo3 expression


© NASA Goddard Space Flight Center - https://www.flickr.com/photos/gsfc/15284914831

Demo

Mapping Evaluation

Naive approach

- Empty semantic candidates expression
  -> Sirius looks for candidates into all loaded semantic/domain model
- **eAllContents()** on each domain resource content
- **Not efficient**
- **No control on the displayed elements**

Diagram elements computation

From the **element** to refresh (and its description/mapping):

- Get **available mappings** to refresh
  - activated Viewpoints, activated Layers
  - children mappings + reused mappings

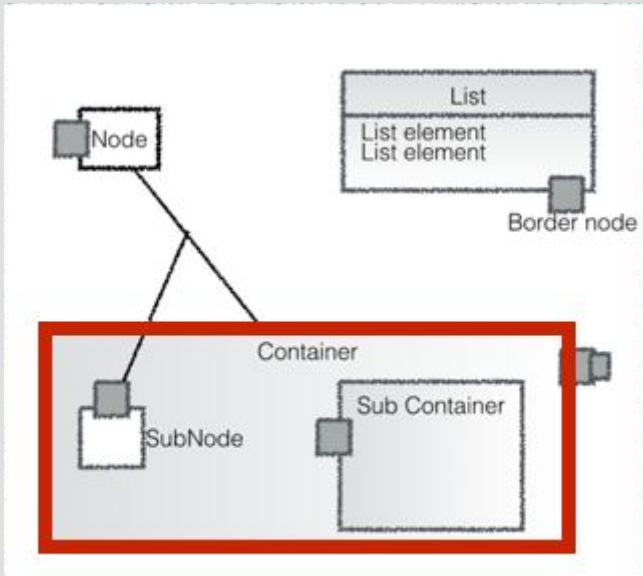


Diagram elements computation

From the **element** to refresh (and its description/mapping):

- For **each** mapping found
  - Evaluation of the **semantic candidates expression** from the current domain element
    (or eAllContents() on each domain resource if empty)
  - Filter with the specified **domain class**
  - On **each** candidate, evaluate the **precondition**
  - Create the diagram element, assign a style

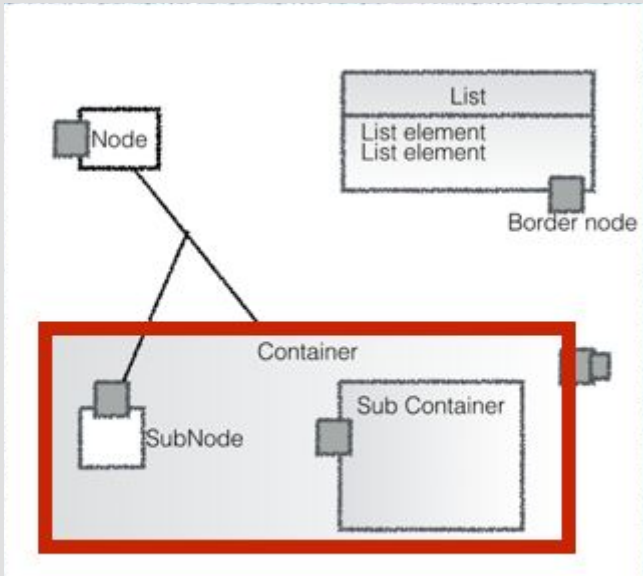


Diagram elements computation

Worst conditions

empty semantic candidates

+ big models

+ many (sub) mappings

+ many complex precondition expressions

⇒ **Poor performances**

Diagram elements computation

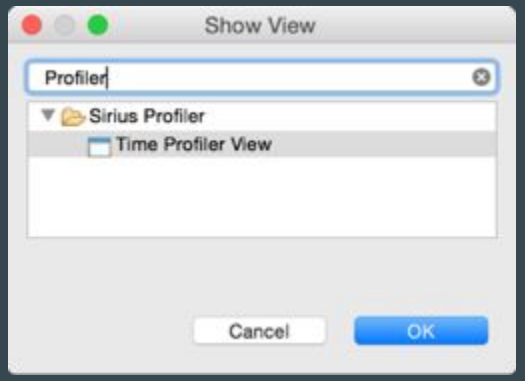
Your role

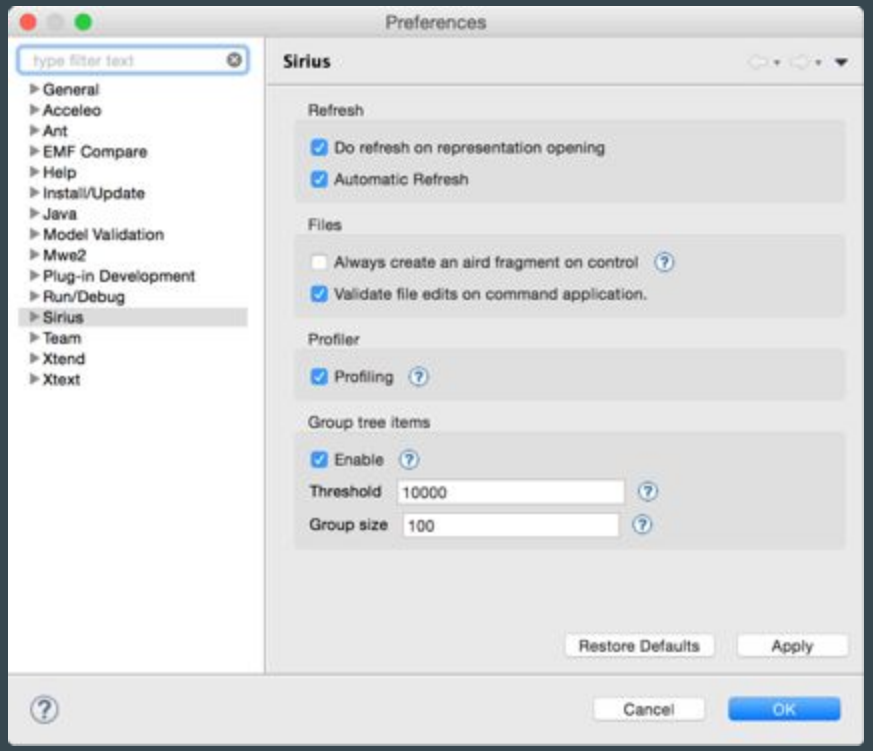
Try as much as possible to write efficient semantic candidates expression:

- Avoid **empty semantic candidates expression** and **eAllContents** when possible
- Follow the structural features defined in the meta model
- Use the **inverse cross references** to look for elements with a reference to another element.
  - **eInverse(Type)** in AQL and Acceleo3
  - access to the **ECrossReferenceAdapter** from a Java service
- Use the **specialized** interpreters when possible (**var:** / **service:** / **feature:**)
- Try to integrate your **precondition** in your **semantic candidate expression**

```
aql: mainExpression -> select( e | e.precondition)
```

Sirius Profiler

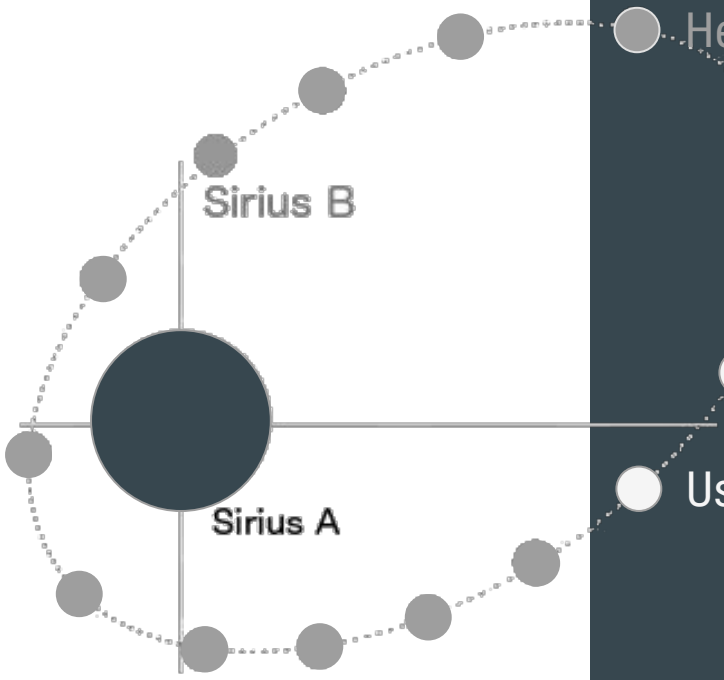




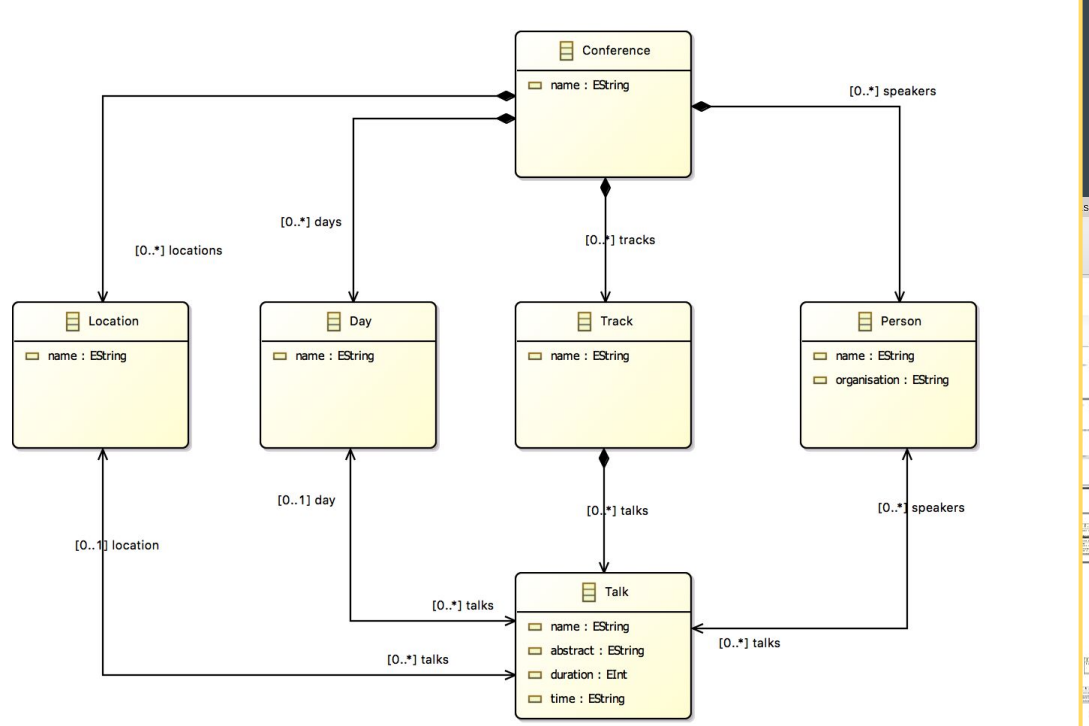
Time Profiler View — Reinit profiler | Refresh View | Print to console

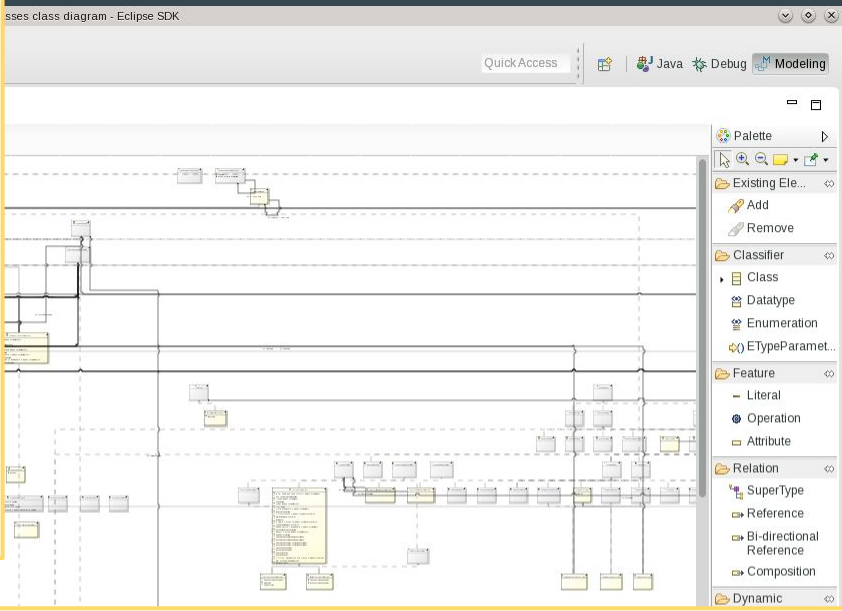
Task Category	Task Name	Time (ms)	Time (hh:mm:ss,ms)	Occurences	Minimum
Acceleo	feature:eOperations	0	0:0:0,0	36	0
Acceleo	service:getVisibleAnnotations(diagram)	2	0:0:0,2	12	0
Other	Other	22	0:0:0,22	0	1
DDiagram	Get edge's candidates	18	0:0:0,18	24	1
DDiagram	Compute edge source/target views	5	0:0:0,5	48	0
DDiagram	Get edge's candidates	13	0:0:0,13	24	1
Acceleo	feature:eType	0	0:0:0,0	12	0
Acceleo	service:eContainerEContainer	0	0:0:0,0	12	0
Acceleo	feature:eContainer	0	0:0:0,0	12	0
Acceleo	feature:eSuperTypes	0	0:0:0,0	18	0
Acceleo	Check precondition expressions	1	0:0:0,1	24	0
DDiagram	Get node's candidates	6	0:0:0,6	12	0
Other	Other	6	0:0:0,6	0	1
DDiagram	Updating all edges	37	0:0:0,37	24	0
Other	Other	141	0:0:0,141	0	20

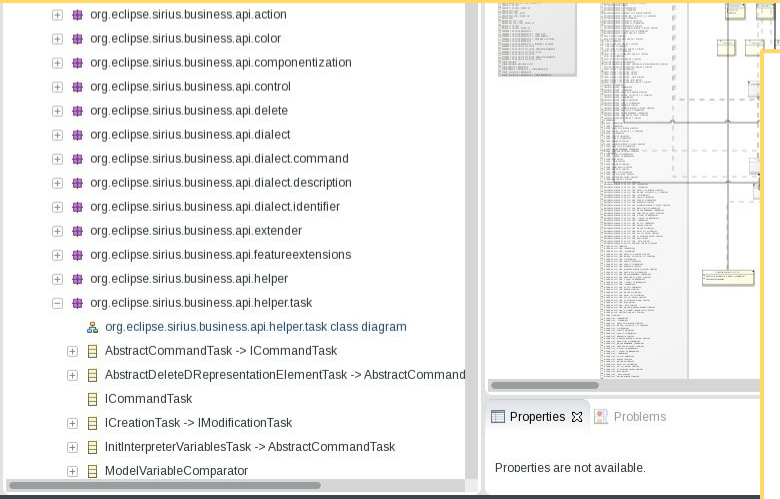
Outline

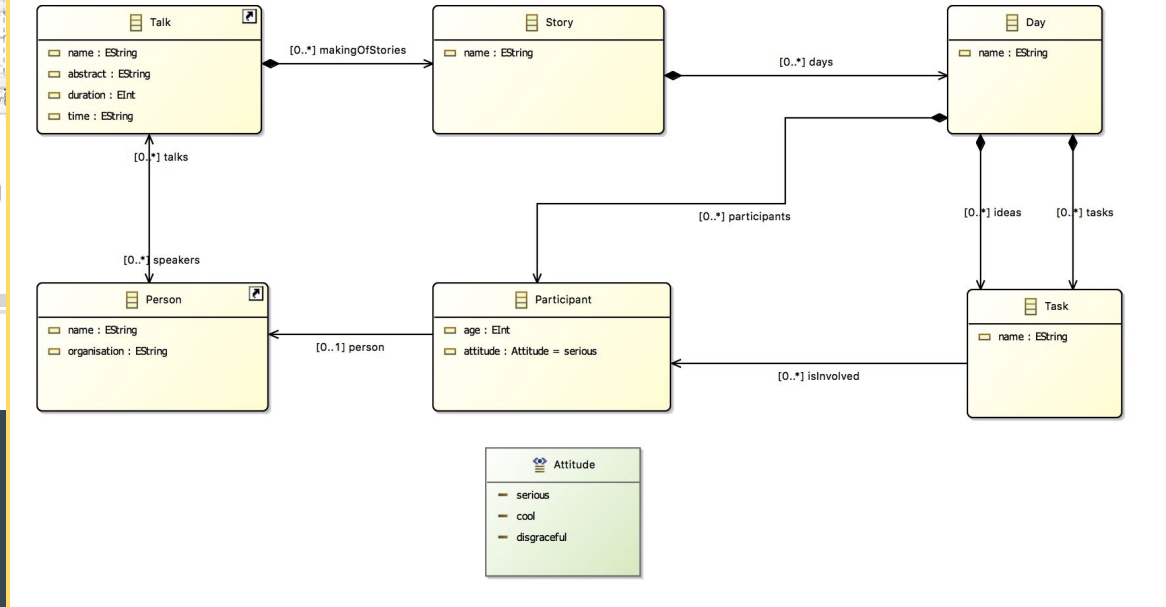


- Help Sirius find the elements to display
- Synchronization options and advanced tools
- Additional mappings and tools contribution
- Style and color customization
- Use the best query language for the task

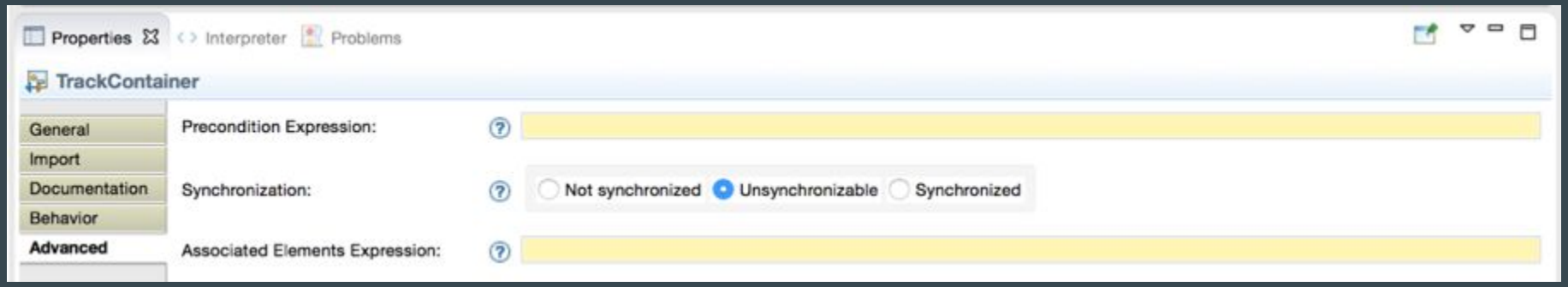

Conference
name : EString
[0..*] speakers
[0..*] days
[0..*] tracks
[0..*] locations
Location
name : EString
Day
name : EString
Track
name : EString
Person
name : EString
organisation : EString
[0..1] day
[0..*] talks
[0..*] speakers
[0..1] location
[0..*] talks
[0..*] talks
[0..*] talks
Talk
name : EString
abstract : EString
duration : EInt
time : EString

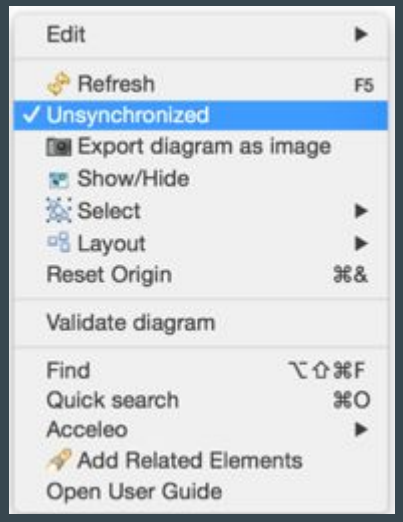

sses class diagram - Eclipse SDK
Quick Access
Java
Debug
Modeling
Palette
Existing Ele...
Add
Remove
Classifier
Class
Datatype
Enumeration
ETypeParamet...
Feature
Literal
Operation
Attribute
Relation
SuperType
Reference
Bi-directional Reference
Composition
Dynamic

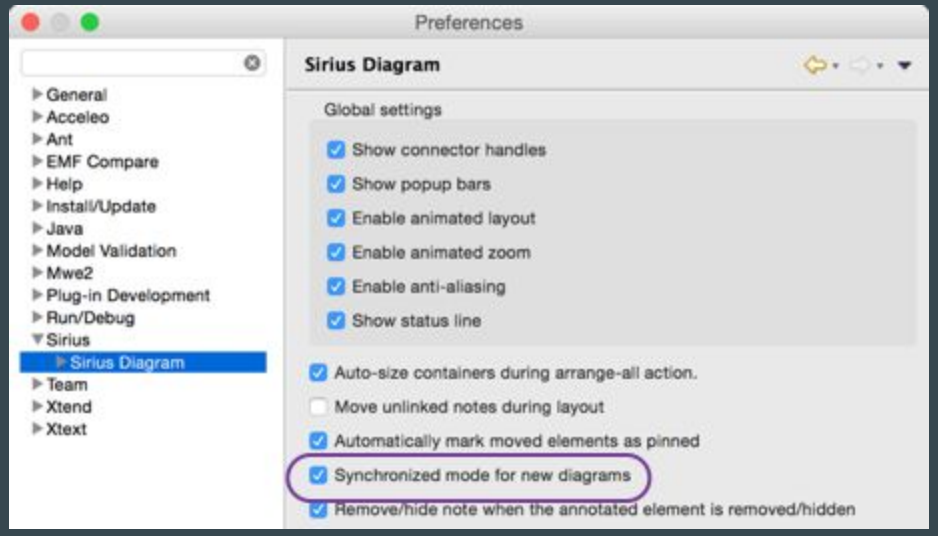

org.eclipse.sirius.business.api.action
org.eclipse.sirius.business.api.color
org.eclipse.sirius.business.api.componentization
org.eclipse.sirius.business.api.control
org.eclipse.sirius.business.api.delete
org.eclipse.sirius.business.api.dialect
org.eclipse.sirius.business.api.dialect.command
org.eclipse.sirius.business.api.dialect.description
org.eclipse.sirius.business.api.dialect.identifier
org.eclipse.sirius.business.api.extender
org.eclipse.sirius.business.api.featureextensions
org.eclipse.sirius.business.api.helper
org.eclipse.sirius.business.api.helper.task
org.eclipse.sirius.business.api.helper.task class diagram
AbstractCommandTask -> ICommandTask
AbstractDeleteDRepresentationElementTask -> AbstractCommand
ICommandTask
ICreationTask -> IModificationTask
InitInterpreterVariablesTask -> AbstractCommandTask
ModelVariableComparator
Properties
Problems
Properties are not available.


Talk
name : EString
abstract : EString
duration : EInt
time : EString
[0..*] makingOfStories
Story
name : EString
[0..*] days
Day
name : EString
[0..*] talks
[0..*] participants
[0..*] ideas
[0..*] tasks
[0..*] speakers
Person
name : EString
organisation : EString
[0..1] person
Participant
age : EInt
attitude : Attitude = serious
[0..*] isInvolved
Task
name : EString
Attitude
serious
cool
disgraceful

Mapping synchronization







Mapping synchronization

- **Synchronized** mapping: Sirius looks for mapping candidates
- **Unsynchronized** mapping: Sirius refreshes styles and sub elements.

- Allows to create **contextual diagrams**:
  - User controls the elements he wants to see on his diagram
  - Sirius does not create elements for non-synchronized mappings
  - Delete from diagram is enabled

Mapping synchronization

- Specifier must create some **‘insertion’** tools
  - Selection Wizards
  - Drop tools (from Model Explorer)
  - Double clics
  - Menus

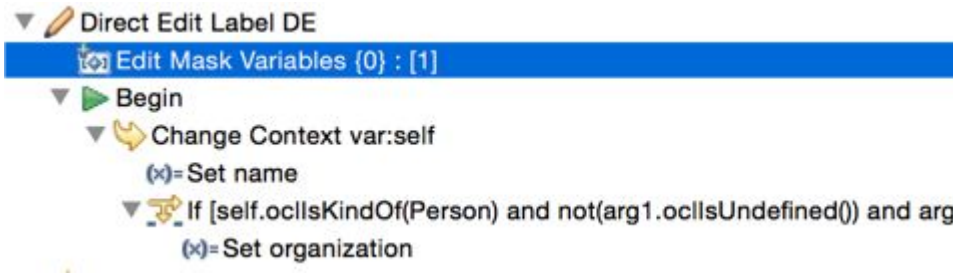
- Mappings of **edge**, **border nodes**, **list elements** often put as **synchronized**

Advanced tools:

Easy edit mask creation

- **{0} : {1}**
- split user text into **String** variables

Direct edit (F2)



Advanced tools:

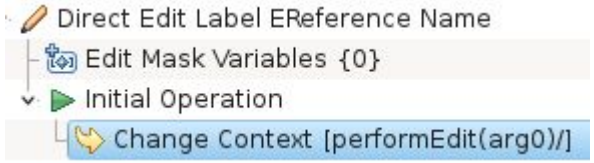
Java services can be used to do more.

Ecore Tools:

direct edit of **EStructuralFeatures** (nodes/edges)

- « Something » => change name of feature
- «:SomeType » => only change the eType
- «1» => only set cardinality to 1..x
- « * » => only set cardinality to x..*
- « /Something » => make the feature derived
- « = something » => set the default value literal
- [...]

Java service



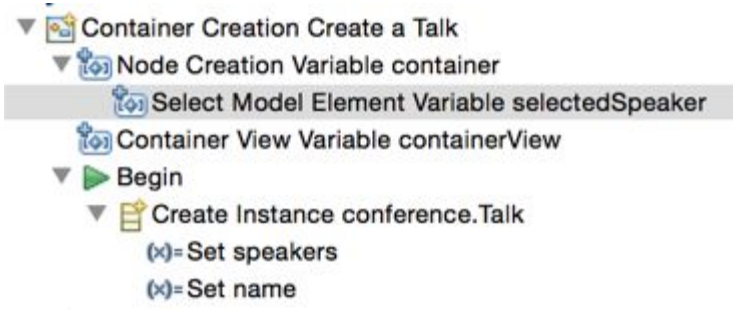

```
public EReference performEdit(EReference ref, String editString) {
    if ("0".equals(editString.trim())) {
        ref.setLowerBound(0);
    } else if ("1".equals(editString.trim())) {
        ref.setLowerBound(1);
    } else if (CARDINALITY_UNBOUNDED.equals(editString.trim())) {
        ref.setUpperBound(-1);
    } else if (CARDINALITY_UNBOUNDED_ALTERNATIVE.equals(editString.trim())) {
        ref.setUpperBound(-1);
    } else {
        editName(ref, editString);
        editCardinality(ref, editString);
    }
    return ref;
}
```

Advanced tools:

Displays a selection dialog
when the user execute a tool

List or tree
Single / Multiple result

Element select variable

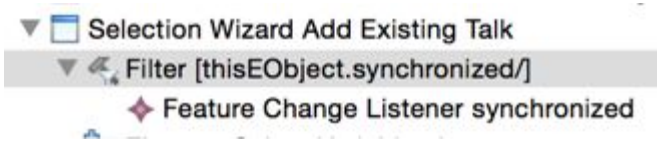


Advanced tools:

To control the visibility of a palette tool

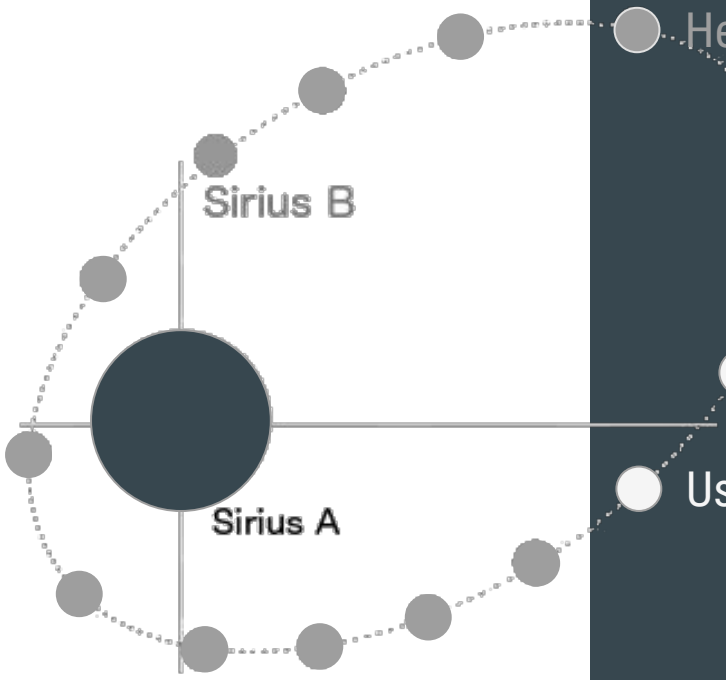
Reacts to model changes
(Sirius or semantic)

Filter listener



Demo

Outline



- Help Sirius find the elements to display
- Synchronization options and advanced tools
- Additional mappings and tools contribution
- Style and color customization
- Use the best query language for the task

Provide mappings and tools

Viewpoint Specification Project

- Ready to deploy Eclipse plugin

Provide mappings and tools

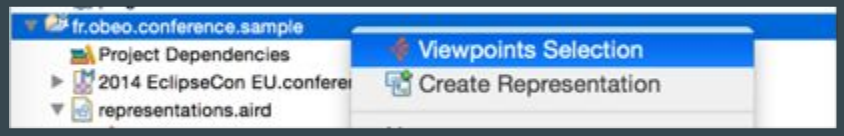
Viewpoint Specification Model

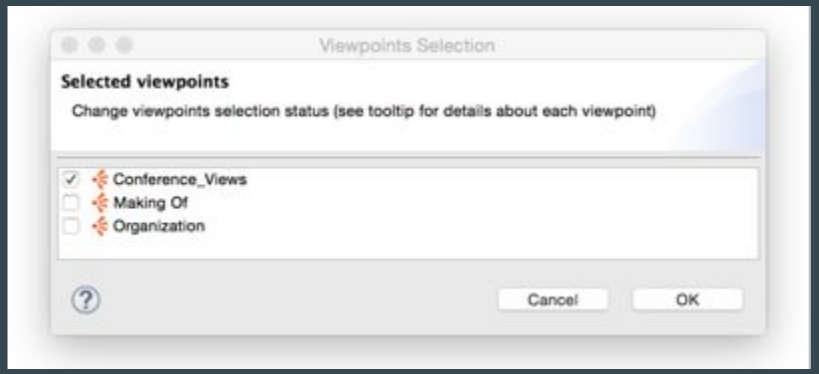
- 1..* per **Viewpoint Specification Project**
- EMF model, can have links to other **VSM**
- Possibility to extends/complete **VSM** defined in other plugins

Provide mappings and tools

Viewpoint

- Declares **Diagram** / **Table** / **Tree description**
- but also **Diagram Extension**
- Activation controlled by the user





Provide mappings and tools

Diagram Description

- 1 default **Layer**
- 0..* additional Layers

Diagram Extension Description

- references a **diagram description** (defined anywhere)
- provides additional Layers

Provide mappings and tools

Layer

- **optional**?
- **active per default**?
- contains top level mappings and tool section

- activation controlled by the user if optional

Provide mappings and tools

Node / Container / Edge mapping import

- to spezialize mappings
- provide new styles / children mappings

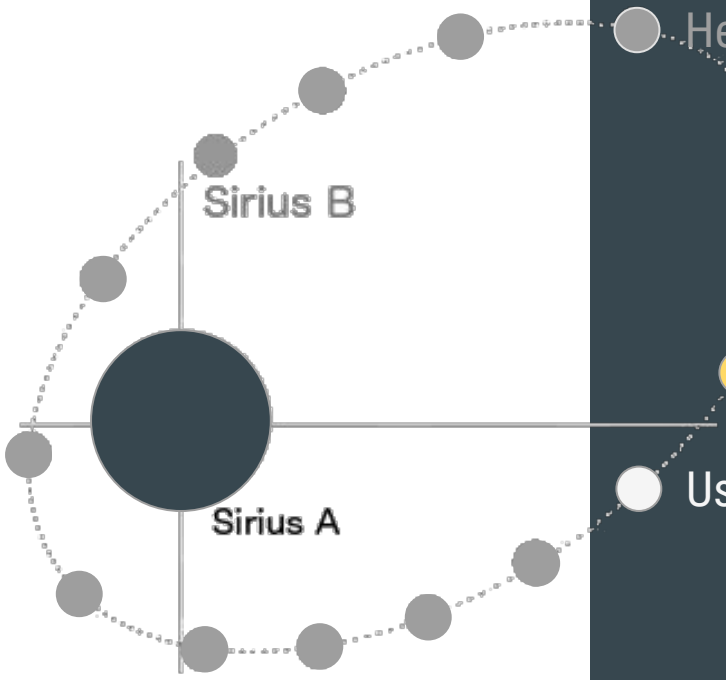
Provide mappings and tools

Tool Section

- contains other tool sections
- declares or reuses tools

Demo

Outline

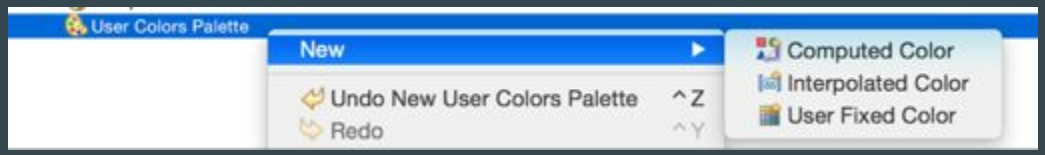


- Help Sirius find the elements to display
- Synchronization options and advanced tools
- Additional mappings and tools contribution
- Style and color customization
- Use the best query language for the task

User colors

Predefined colors

User Color Palette



- **User fixed color**: RGB, System color chooser
- **Computed Color**: interpreted expression to compute R, G, B
- **Interpolated Color**
  - Define several color steps (value/color)
  - Expression to compute a value from the element to decorate

Conditional Styles

- Available for every kind of mapping
- **0..*** conditional style
- Each conditional style contains a **different style**

Conditional Styles

Node styles

				
Basic Shape Square	**Basic Shape Stroke**	**Basic Shape Triangle**	**Basic Shape Dot**	**Basic Shape Ring**

						
Square	**Lozenge**	**Ellipse**	**Note**	**Gauge**	**Image**	**Custom**

Conditional Styles

Container styles

Gradient	Image

Conditional Styles

Edge styles

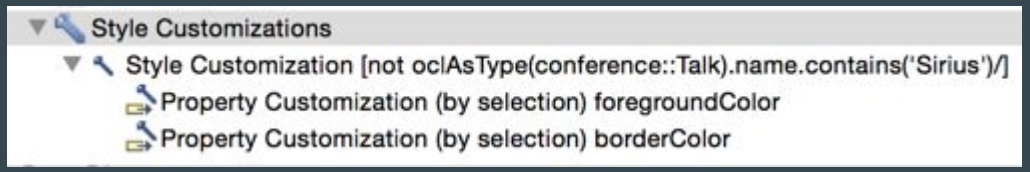
- Routing style (oblique, manhattan, tree)
- Line style
- Source / Target arrows
- Begin / Center / End labels

Conditional Styles

- Precondition must be **exclusive**
- Sirius takes the first whose precondition evaluation returns true.

- The **‘default’** style is taken if no conditional style can be applied

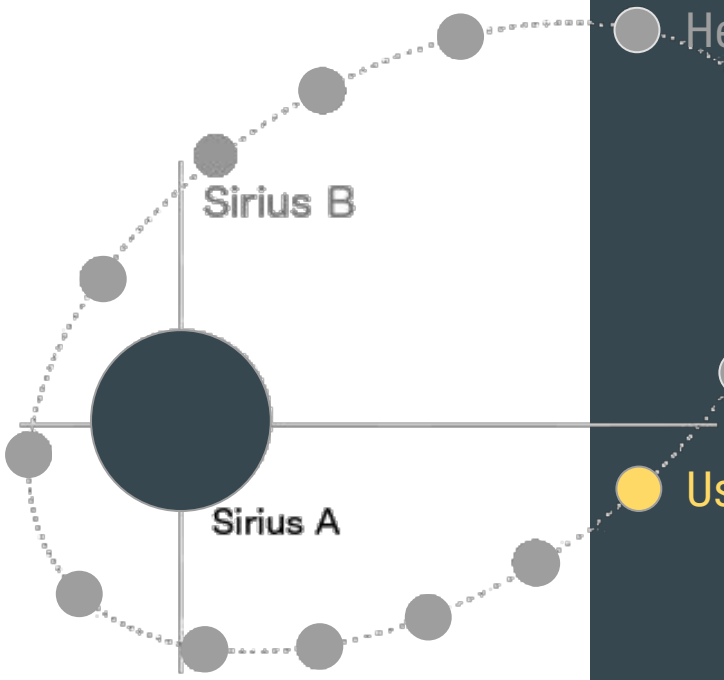
Style Customizations



- Defined in a Layer
- Style Customization has a precondition
- More **fine grained customization** (than the Conditional Styles)
- Property Customization
  - target one **EStructuralFeature** of the Sirius style descriptions
  - applied on all **styles** or selected ones

Demo

Outline



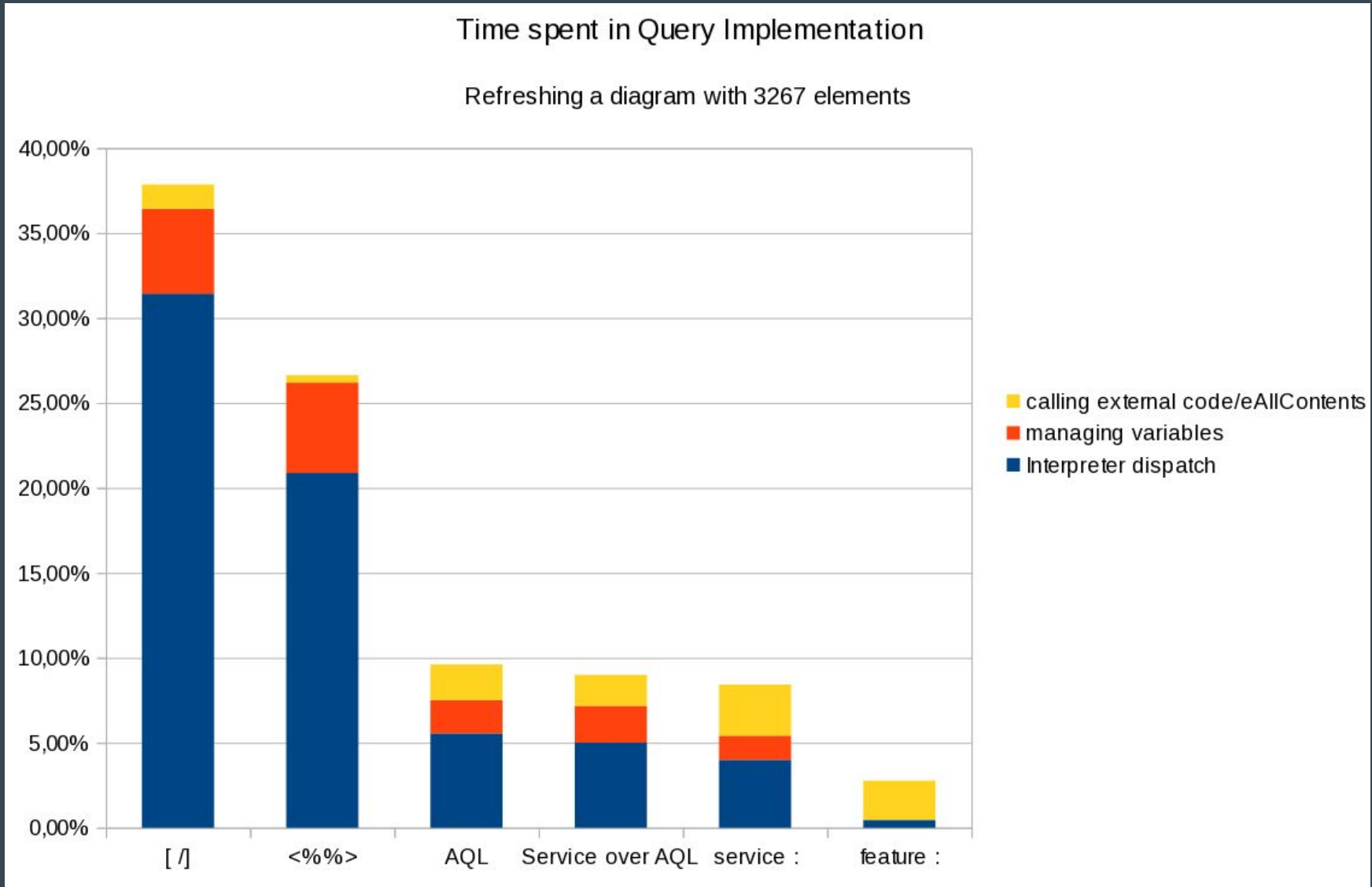
- Help Sirius find the elements to display
- Synchronization options and advanced tools
- Additional mappings and tools contribution
- Style and color customization
- Use the best query language for the task

Provided interpreters

- **var:** direct access to Sirius variables
- **feature:** direct access to the named features of the current element
  (and EMF pseudo-features)
- **service:** direct call of a Java method
  (that follows some naming conventions, see documentation)
- **[/]**: Acceleo3 expression
- **aql:** Acceleo Query Language
  (introduced with Sirius 3.0, **recommended** since 3.1)

Note: extensible through extension point

Provided interpreters



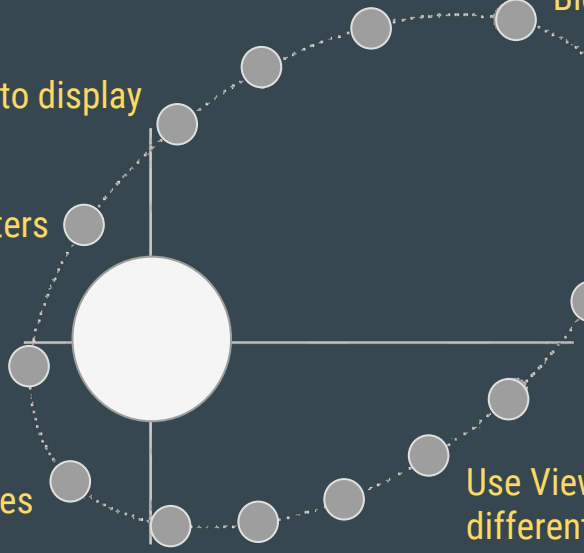
Provided interpreters

AQL

- **stronger type information** than Acceleo3 - allows stronger type analysis
- implementation specifically tailored for the **Sirius** use case
- complex or custom logic: **Java Services**
- **predicatable** ordering and performance overhad
- simple for querying EMF models
- evaluation: **fast** and collect errors
- validation: **strong** and precise

Recommended query language for Sirius 3.1.0

Takeaways



Big models? Remember the contextual diagrams

Help the user with advanced tools

Conditional styles completely change the appearance of an element

Style customizations allow a more fine grained customization

Use Viewpoint, Diagram Extensions, Layers to deal with different aspects in your modelers

Use inverse cross references

Think about your models topology

Use AQL and the specific interpreters

Help Sirius find the elements to display

Performances depends on your .odesign specification

Measure, Improve, Repeat

Research projects
Funding
Sys2Soft
DGE
DIRECTION GÉNÉRALE DES ENTREPRISES

Thank you!

Maxime Porhel
http://mporhel.github.io/slides/
maxime.porhel@obeo.fr
@mporhel