

Eclipse Sirius Demonstration

An Open Source Technology to Create Custom Modeling Workbenches

Frédéric Madiot

Obeo
Nantes, France
frederic.madiot@obeo.fr

Marc Paganelli

Obeo
Montréal, Canada
marc.paganelli@obeo.fr

This demonstration will present Sirius[1], an Eclipse project which allows you to easily create your own graphical modeling workbench by leveraging the Eclipse Modeling technologies, including EMF[2] and GMF[3].

<https://www.youtube.com/watch?v=rUqSaxu6SqM>

I. BACKGROUND

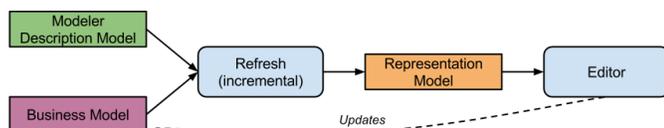
Sirius has been created by Obeo and Thales to provide a generic workbench for model-based architecture engineering that could be easily tailored to fit specific needs.

Based on a viewpoint approach, Sirius makes it possible to equip teams who have to deal with complex architectures on specific domains.

II. PROPOSED APPROACH

A modeling workbench created with Sirius is composed of a set of Eclipse editors (diagrams, tables and trees) which allow the users to create, edit and visualize EMF models.

The innovative approach lies on declarative descriptions. All shape characteristics and behaviors can be easily configured with a minimum technical knowledge. The editors are defined by a model which defines the complete structure of the modeling workbench, its behavior and all the edition and navigation tools. This description is dynamically interpreted using GMF runtime to materialize the workbench within the Eclipse IDE.



In contrast of GMF tooling approach, no code generation is involved, the specifier of the workbench can have instant feedback while adapting the description. Once completed, the modeling workbench can be deployed as a standard Eclipse

plugin. Thanks to this short feedback loop a workbench or its specialization can be created in a matter of hours

In addition, Sirius runtime natively provides mechanisms for managing models complexity:

- **layers**: it is possible to allocate specific graphical elements to layers that can be activated/deactivated on demand by the user
- **filters**: to automatically show or hide elements of a diagram with a condition that determines which elements remain visible.
- **styles customization** : to change graphical properties of diagram elements (color, label, size, ...) depending on conditions.
- **validation rules** : allow the user to evaluate the quality of a model.
- **quick fixes**: to allow the user to solve the problems automatically.
- **navigation tools**: to facilitate the possibility for the user to drill down the model through simple diagrams

For supporting specific need for customization, Sirius is extensible in many ways, notably by providing new kinds of representations, new query languages and by being able to call Java code to interact with Eclipse or any other system.

III. RESULT

Available as Open Source, Sirius is integrated into annual versions of the Eclipse platform thanks to compliance with very strict development standards and a stringent development process. It guarantees the quality and transparency that have built the reputation of this worldwide distributed platform.

Since the earliest contributions, the Sirius team has also worked hard to ensure a broad community of users signs up to this technology.

Sirius lies at the heart the "Capella[4]" Systems Engineering workbench and has been extensively deployed in operational

units for the aerospace, transport, energy and communication sectors.

Sirius is also the cornerstone of the management application design workbench of the French Ministry of Defense and underpins a technology used to create satellite applications for the European Space Agency.

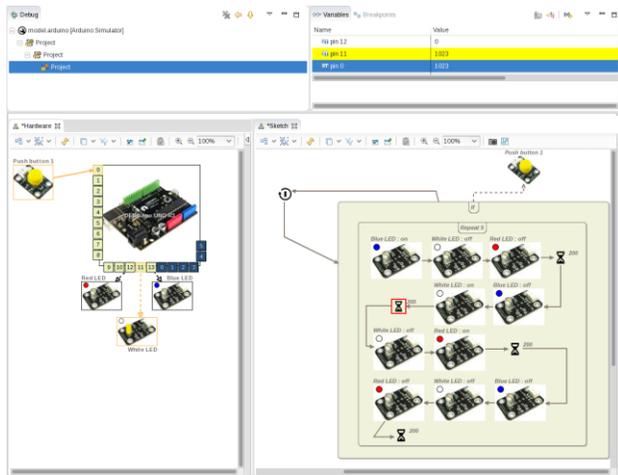
Many other usages of this technology are listed on the Sirius's website : <http://www.eclipse.org/sirius/gallery.html>

IV. RELATED WORK

A. Sirius Animator

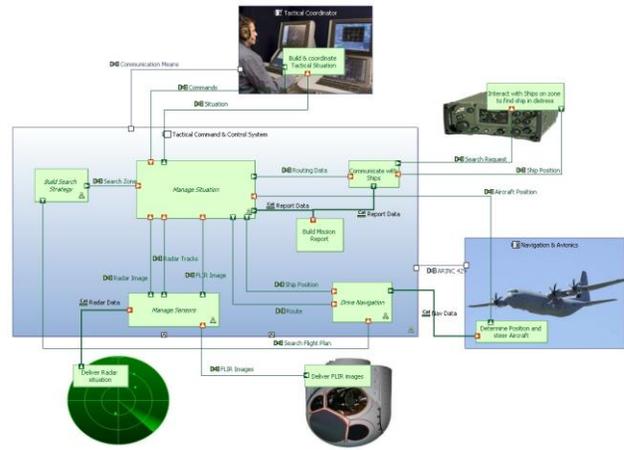
Sirius Animator[5] is an extension to Sirius developed in the GEMOC[6] project. It brings execution capabilities for your DSL or any EMF based models in a quick and flexible way.

The Sirius Animator has been used in an experimental branch of Arduino Designer to bring instant feedback on a given design without having to compile and deploy the program on the Arduino target. Plugins integrating UML Designer, an UML engine and the Eclipse Debug API have also been prototyped.



B. Capella

Capella is a model-based engineering solution that has been successfully deployed in a wide variety of industrial contexts. Based on a graphical modelling workbench, it provides systems, software and hardware architects with rich methodological guidance relying on Arcadia, a comprehensive model-based engineering method.

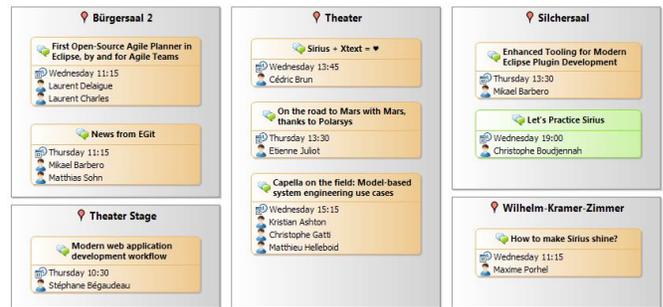


V. DEMONSTRATION

The Sirius demonstration will illustrate how to create a new kind of diagram for an existing DSL (Domain Specific Language) defined with Ecore Tools[7].

The example will be the MODELS conference itself! Starting from an Ecore model defining the concepts of Conference, Track, Talk, Location, Person, we will create a graphical editor to display the models describing MODELS 2015. The talks and the speakers will be represented with boxes and edges between them. In this editor we will also add tools to create new talks.

We will then show more advanced diagrams showing different viewpoints over the conference (by day, by track, by room) and a table presenting the whole program.



Finally, we will briefly show real-life tools created with Sirius : UML Designer[8], BPMN Designer[9], Arduino Designer[10] and Capella.

VI. REFERENCES

[1] Sirius website: <http://www.eclipse.org/sirius/>
 [2] EMF website: <https://eclipse.org/modeling/emf/>
 [3] GMF website: <http://www.eclipse.org/modeling/gmf/>

[4] Capella website: <https://www.polarsys.org/capella>

[5] Sirius Lab page: <http://www.eclipse.org/sirius/lab.html>

[6] GEMOC project : <http://gemoc.org/>

[7] Ecore Tools website: <http://www.eclipse.org/ecoretools/>

[8] UML Designer website: <http://www.uml designer.org/>

[9] BPMN Designer website:
<http://marketplace.obeonetwork.com/module/bpmn>

[10] Arduino Designer website:
<https://github.com/mbats/arduino>