

Improvement of the OSRA SCM Model Editor

--

TEC-ED & TEC-SW Final Presentation Days

6-7 December 2016

--

Mr. Andreas Jung (ESA/ESTEC)

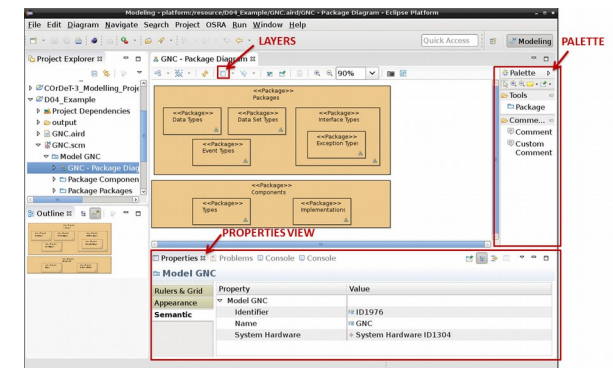
Mr. Goulwen Le Fur (Obeo)



Introduction

OSRA and component model nothing without a tool

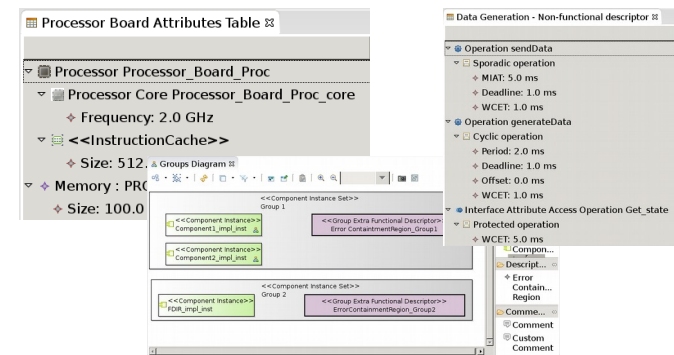
- ESA's R&D activities (mainly COrDeT) developed the
 - OBSW Reference Architecture (OSRA) and the
 - Space Component (meta-)Model (SCM)
- Several documents produced (Training material, soon also SAVOIR documents)
 - However, the possibility to experiment with a tool the component model approach is considered essential for the full understanding of OSRA (and for any tool chain associated to it)
- COrDeT activities prototyped a graphical editor to experiment and test the complete OSRA approach (based on Eclipse and Obeo Designer – tools fitted well with DSL defined in ecore).



Development and improvement of graphical tool

- However, following the COrDeT activities, it was clear that an improved version of the editor, in terms of usability, is needed.

- Prototype was not user-friendly:
 - 12 diagrams with 19 tables
 - a lot of steps to create an initial model
 - related information (tables/diagrams) were not visible at the same time
 - no workflow support



...but it was never the intention of the prototype to be user friendly
– the objective was to experiment and validate the meta-model.

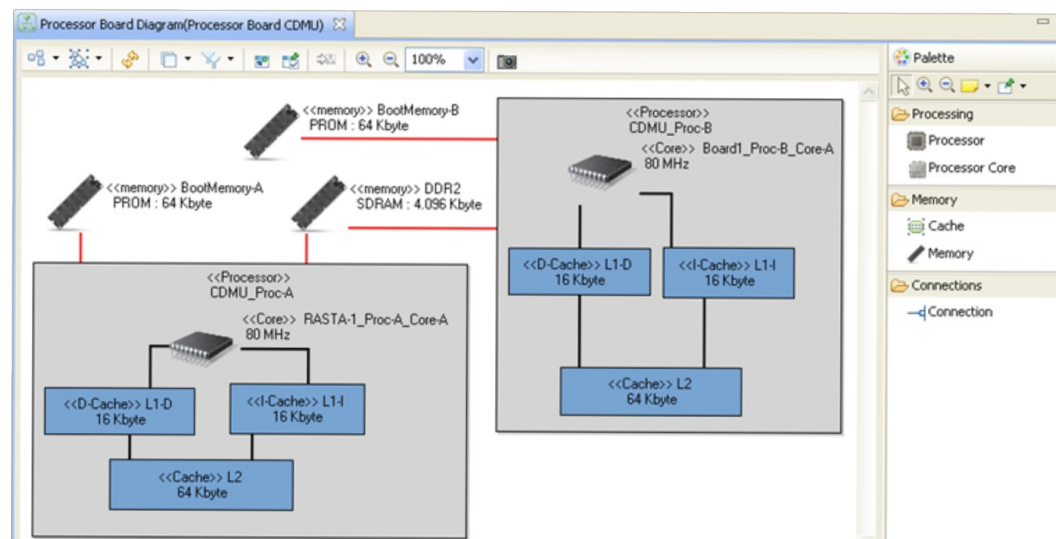
- Hence, ESA target in a small lab investment activity (70kEUR) to produce tool with focus on User Experience (UX)



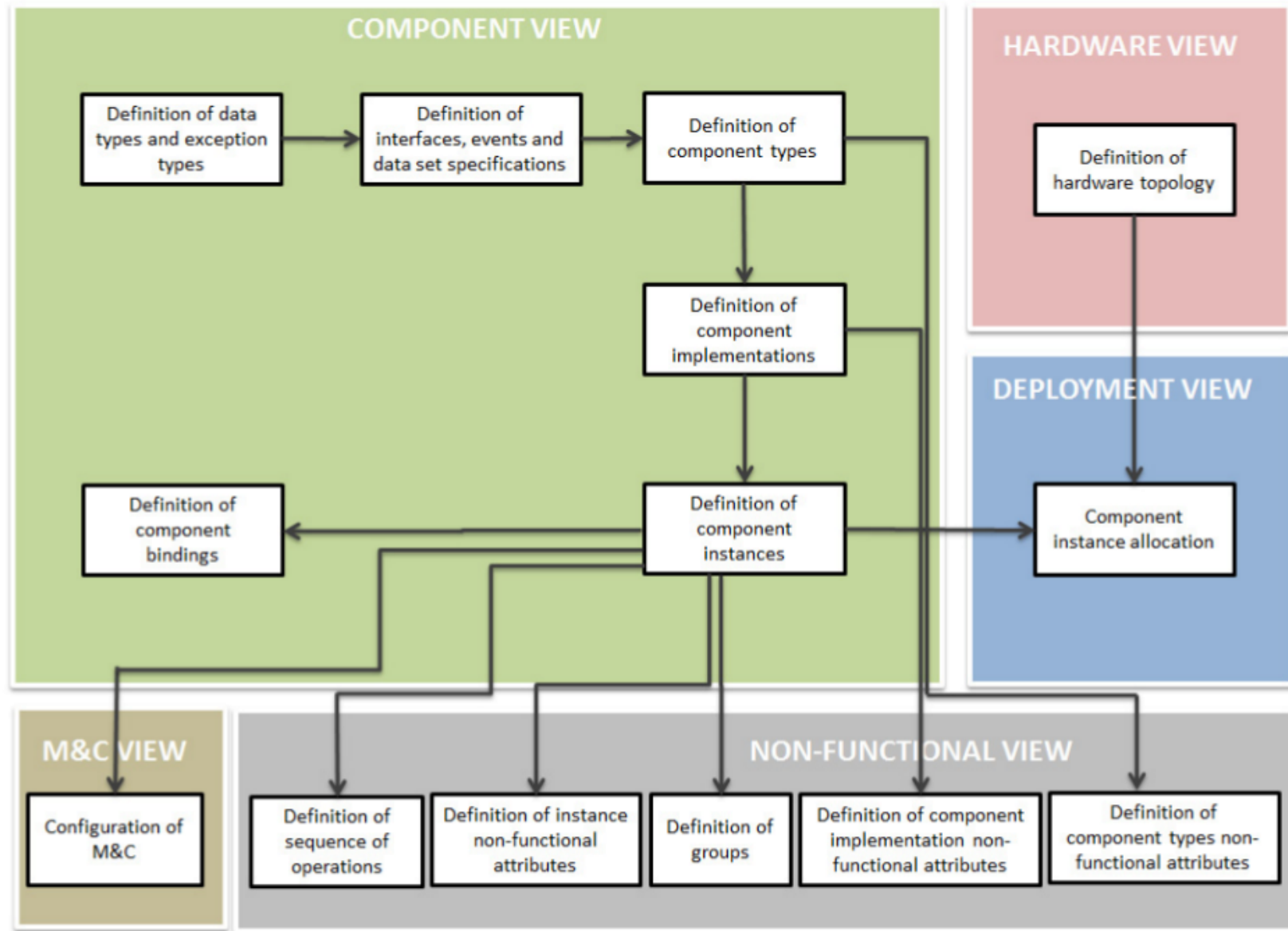
**Context
Of the
Activity**

Back to the start...

- COrDeT was initially developed on Obeo Designer
 - An Eclipse distribution marketed by Obeo
 - Mixing proprietary and Open Source frameworks
 - Including a technology to define graphical modelers



A modeling tool with potential



A technical transition...

- In 2013, Obeo releases its graphical technology in Open Source via an Eclipse project called Sirius
- The OSRA Editor Improvement activity was restarted on the Eclipse Sirius project
 - To ensure long time support due to Open Source
- The transition was easy due to the backward compatibility between Sirius and Obeo Designer

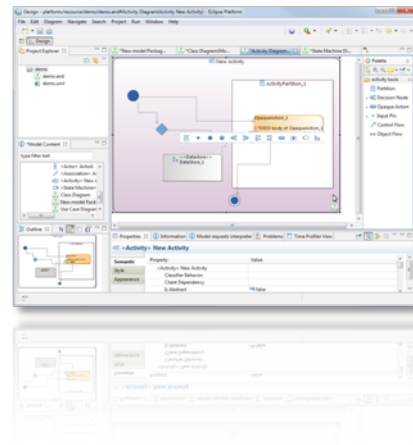


Let's be concrete...

Sirius



Business Domain



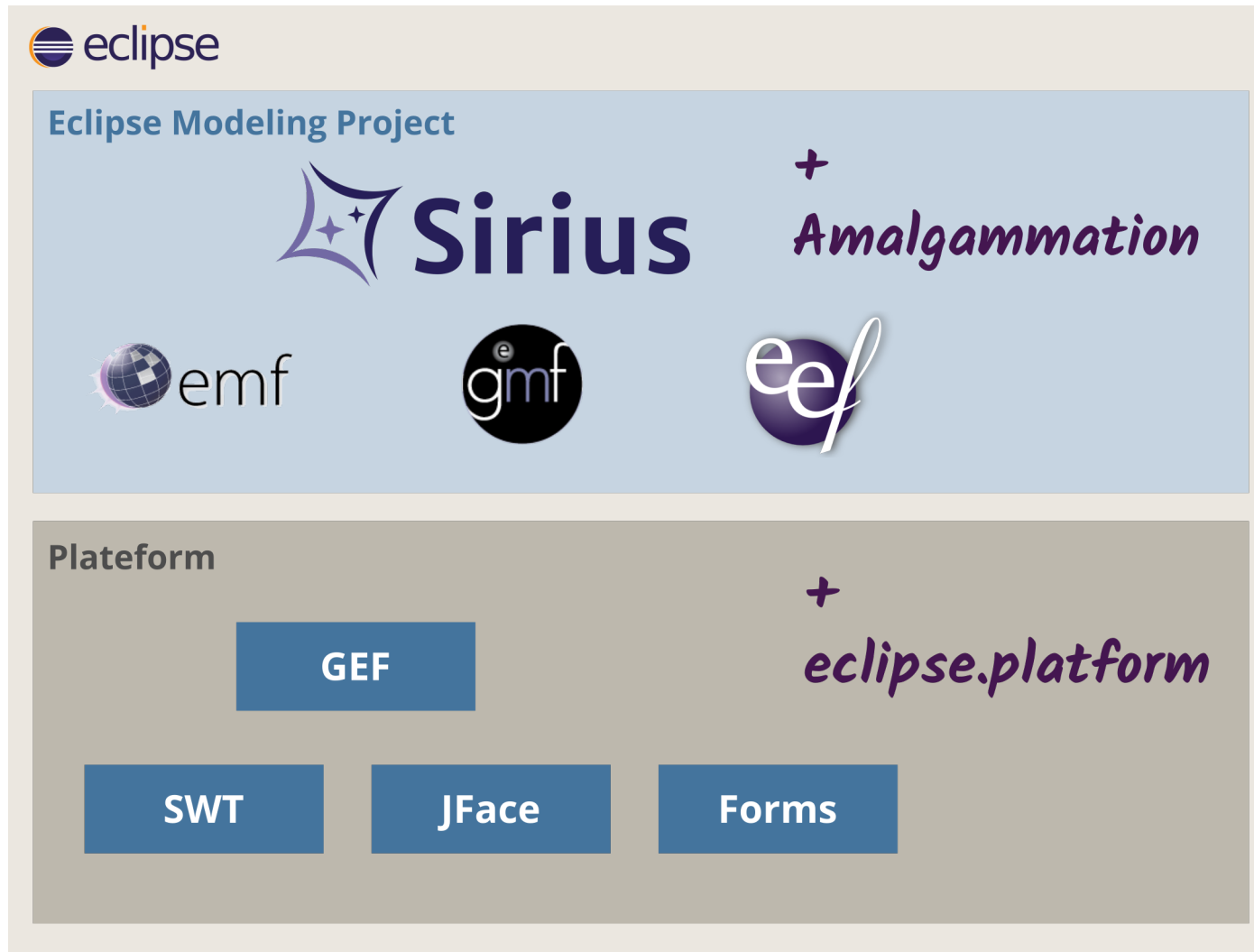
Modeling Workbench



Users

**Easily and efficiently create
Custom modeling workbenches**

Graphical modelers, but not only...



In action...

The screenshot displays the OSRA Expert Eclipse Platform interface. The main window shows a **Component Instance Diagram** with the following elements:

- Control_Type_impl_inst** (Component Instance): Provides **Control_IF** and **Actuator_IF** interfaces. It is connected to **Guidance_Type_impl_inst** via **ChangeOfMode_T_EvEm** and **ChangeOfMode_T_EvRc** events.
- Guidance_Type_impl_inst** (Component Instance): Provides **Guidance_IF** and **Actuator_IF** interfaces. It is connected to **Actuator_Type_impl_inst** via **Actuator_IF** and to **Navigation_Type_impl_inst** via **Guidance_IF**.
- Actuator_Type_impl_inst** (Device Instance): Provides **Actuator_IF** interface.
- Navigation_Type_impl_inst** (Component Instance): Provides **Guidance_IF** and **DataAcquisition_IF** interfaces.
- Sensor_Type_impl_inst** (Device Instance): Provides **Sensor_IF** and **Attitude_T** interfaces.

The **OSRA Project Explorer** on the left shows the project structure for **EagleEye**, including **Model EagleEye** and **Package Interfaces, Events and D**.

The **OSRA Context Explorer** at the bottom left shows the **Component Instance Guidance_Type_impl_inst** context, with sections for **Uses...**, **Current element**, and **Is used by...**.

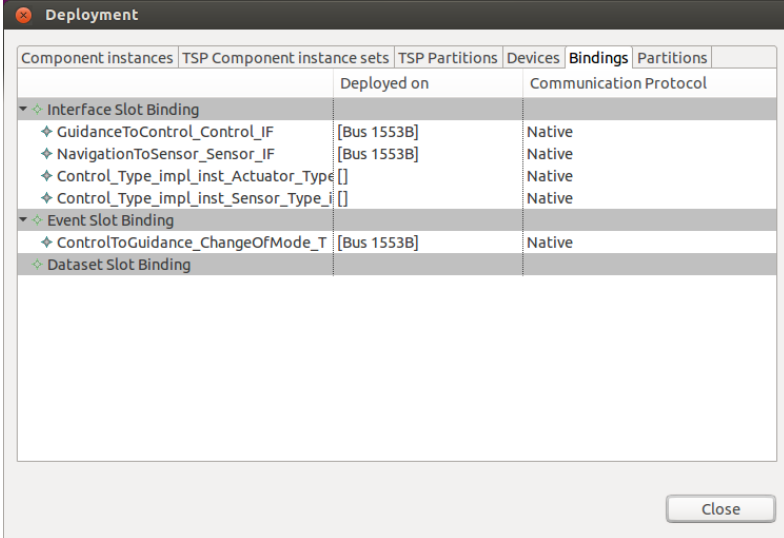
The **Properties** view at the bottom right shows the **Component Instance Guidance_Type_impl_inst** properties, including **OSRA**, **Semantic**, **Style**, and **Appearance** sections.

Good... but not enough

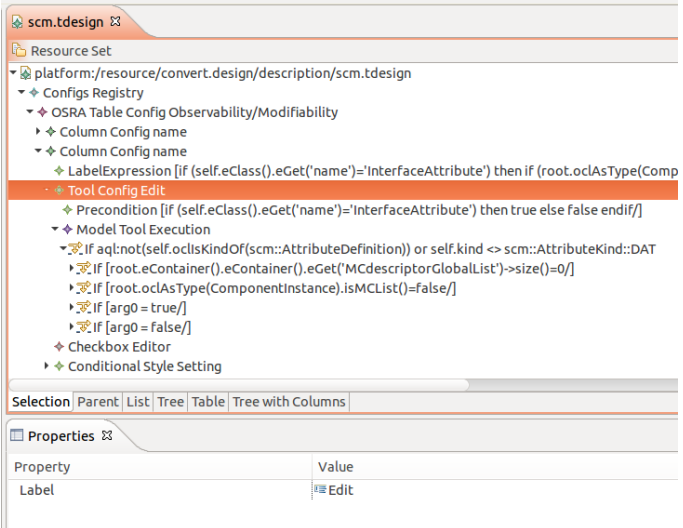
- Most of the features required by the new OSRA editors were covered by the projects listed above
- However, some features weren't offered by any Eclipse project
 - These features were developed during the activity

The OSRA Tables Framework

- An autonomous Table Framework
 - Covering OSRA needs
- Including its own specification mechanism
 - TDesign files
- Reusable for other use cases



Component instances	TSP Component instance sets	TSP Partitions	Devices	Bindings	Partitions
		Deployed on			Communication Protocol
↳ Interface Slot Binding					
↳ GuidanceToControl_Control_IF	[Bus 1553B]			Native	
↳ NavigationToSensor_Sensor_IF	[Bus 1553B]			Native	
↳ Control_Type_impl_inst_Actuator_Type	[]			Native	
↳ Control_Type_impl_inst_Sensor_Type	[]			Native	
↳ Event Slot Binding					
↳ ControlToGuidance_ChangeOfMode_T	[Bus 1553B]			Native	
↳ Dataset Slot Binding					

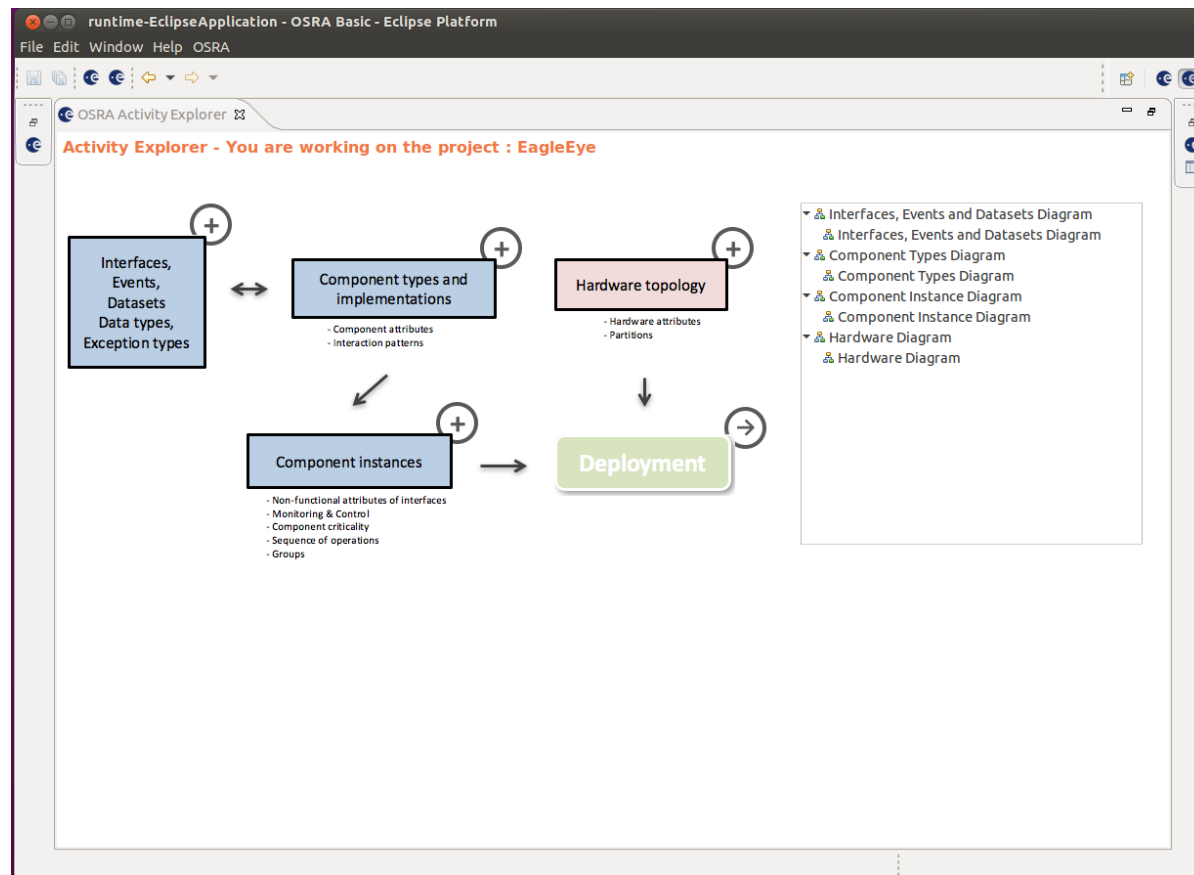


```
scm.tdesign
Resource Set
platform:/resource/convert.design/description/scm.tdesign
  Configs Registry
  OSRA Table Config Observability/Modifiability
  Column Config name
  Column Config name
  LabelExpression [if (self.eClass().eGet('name')='InterfaceAttribute') then if (root.oclAsType(Comp
  Tool Config Edit
  Precondition [if (self.eClass().eGet('name')='InterfaceAttribute') then true else false endif/]
  Model Tool Execution
  If [not (self.oclIsKindOf(scm::AttributeDefinition)) or self.kind <= scm::AttributeKind::DAT
  If [root.eContainer().eContainer().eGet('MCdescriptorGlobalList')->size()=0/]
  If [root.oclAsType(ComponentInstance).isMCList()=false/]
  If [arg0 = true/]
  If [arg0 = false/]
  Checkbox Editor
  Conditional Style Setting
```

Property	Value
Label	Edit

The OSRA Activity Explorer

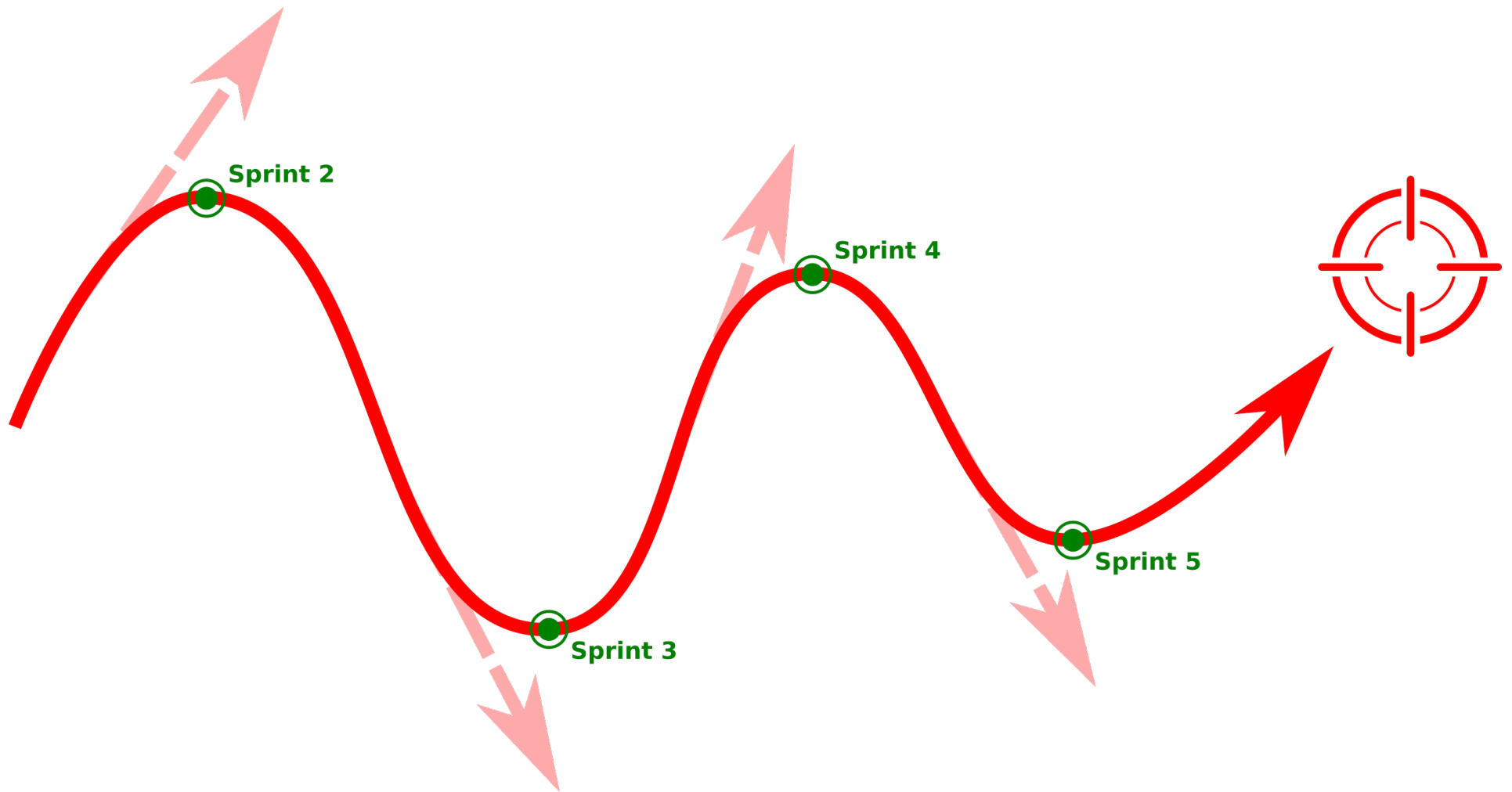
- Dedicated Activity Explorer
 - Inspired from Capella Tooling





Project Organization

A project organized with agility





- Sprints: ~ 2 weeks long
- Structure
 - Progress Meeting
 - Development
 - Tests by ESA Team
 - Feedback before the next PM
- 7 sprints during the activity

What about Obeo in this activity?

- Expertise on the Eclipse/Modeling technologies
 - Guide choices in the large Eclipse ecosystem
- Share of experience on similar projects
 - Support on the workbench conception
- Development
 - Technical lead
 - Work on shared Git repository
 - Work with granularity sufficiently fine to allow the ESA team to check implementation later



Results

The workbench really improved

- A new tool, fully redesigned
 - From a set of Sirius representations to a whole modeling workbench well integrated
- A tool designed for the users
 - With a method
 - Focusing on the UX
- A tool ready for industrial experiments

Complete workbench definition

The screenshot displays the OSRA Expert Eclipse Platform interface. The main window shows a **Component Instance Diagram** with the following elements:

- Control_Type_impl_inst** (Component Instance): Provides **Actuator_IF** and **Control_IF** interfaces.
- Guidance_Type_impl_inst** (Component Instance): Provides **Guidance_IF** and **Control_IF** interfaces.
- Navigation_Type_impl_inst** (Component Instance): Provides **Guidance_IF** and **DataAcquisition_IF** interfaces.
- Actuator_Type_impl_inst** (Device Instance): Provides **Actuator_IF** interface.
- Sensor_Type_impl_inst** (Device Instance): Provides **Sensor_IF** and **Attitude_T** data.

Interactions include **ChangeOfMode_T_EvEm** and **ChangeOfMode_T_EvRc** events between the Control and Guidance components.

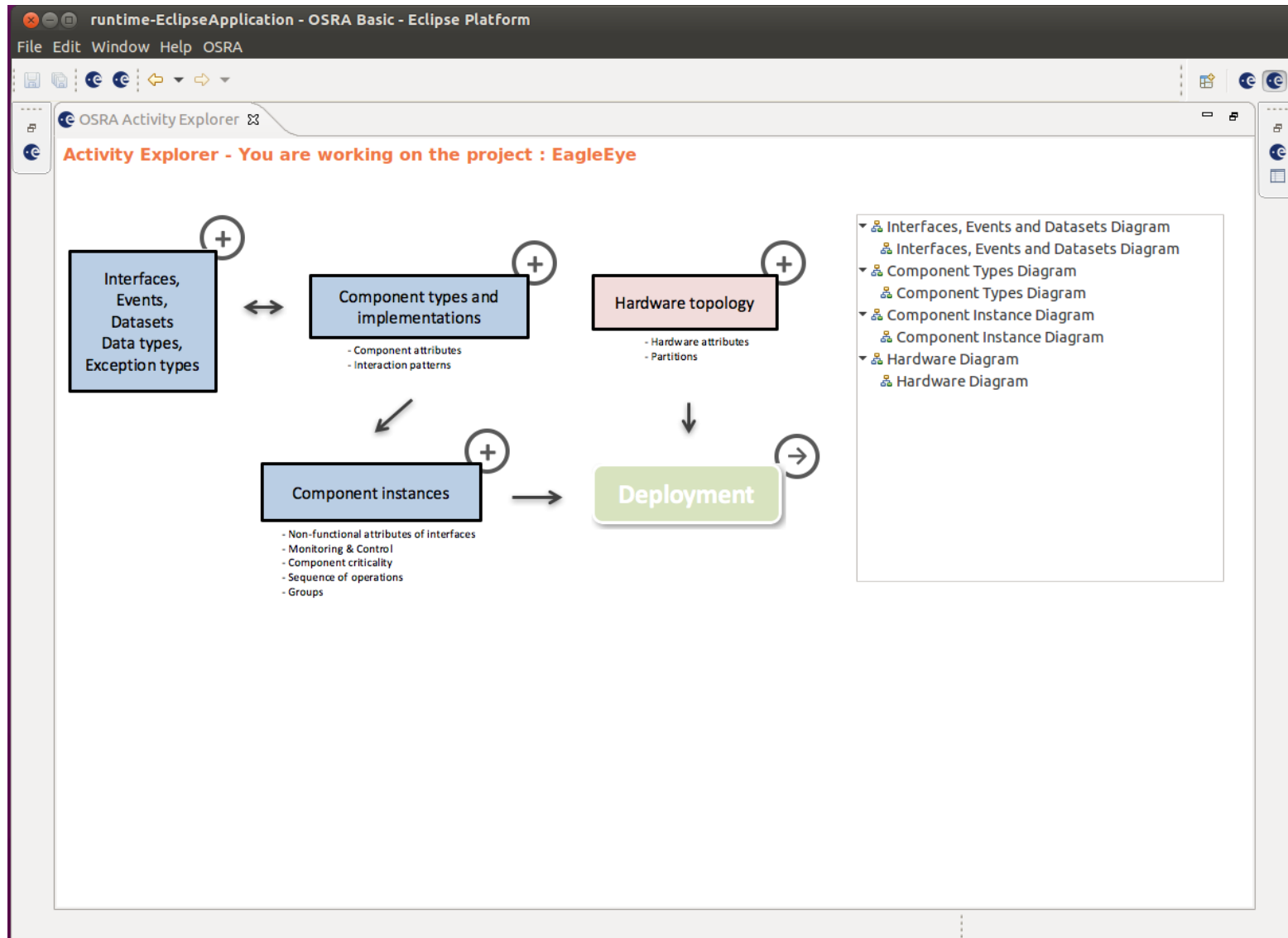
The **OSRA Project Explorer** on the left shows the project structure:

- EagleEye [osra master]
- Project Dependencies
- EagleEye.aird
- EagleEye.scm
- Model EagleEye
 - Package Interfaces, Events and Datasets
 - Package Component Types
 - Component Types Diagram
 - Component Type Navigation_Type
 - Component Type Control_Type
 - Component Type Guidance_Type
 - Package Component Implementations
 - Monitor And Control Descriptor Global List
 - System Deployment
- Component Instance Diagram
 - Slot Binding Deployment Descriptor
 - Deployment Descriptor 618f0c10-86ab-45c3
 - Deployment Descriptor 8e1443a2-a8eb-4ff9
 - Deployment Descriptor 953e5463-90b4-44b1
 - Deployment Descriptor c913b260-4a5b-4c84
 - Deployment Descriptor 569608cb-1026-414e
 - Deployment Descriptor e62bf5ad-9b95-4e73
 - Deployment Descriptor aac95777-2d24-4a3d
 - Component Instance Control_Type_impl_inst
 - Custom Comment TODO: interaction with
 - Provided Interface Slot Control_IF

The **OSRA Context Explorer** at the bottom shows the context for **Component Instance Control_Type_impl_inst**:

Uses...	Current element	Is used by...
<ul style="list-style-type: none">Events<ul style="list-style-type: none">Event ChangeOfMode_TImplementations<ul style="list-style-type: none">Component Implementation ContInterfaces<ul style="list-style-type: none">Interface Actuator_IFOperation command	<ul style="list-style-type: none">Component Instance Control_TypeBindings<ul style="list-style-type: none">Event Slot Binding ControlToGuiInterface Slot Binding Control_TInterface Slot Binding GuidanceTAll Related Diagrams<ul style="list-style-type: none">Component Instance Diagram	<ul style="list-style-type: none">Deployment<ul style="list-style-type: none">Processor Board Processor Board 1

A tool, with a method...



UX focused rework

The screenshot displays the Eclipse IDE interface for OSRA. The top bar shows the application title and the current diagram being edited: "runtime-EclipseApplication - OSRA Expert - platform:/resource/EagleEye/EagleEye.aird/Interfaces, Events and Datasets Diagram - Eclipse Platform".

The **OSRA Project Expl** view on the left shows a tree structure of the project, with "Interface Attribute mode" selected. A red box highlights a button in this view.

The main diagram area shows several UML-like elements: interfaces (Guidance_IF, Control_IF, Sensor_IF), an event (newMode: Mode_T), and various data types (OutOfBounds, FloatArray3_T, Velocity_T, Array Float_T). A context menu is open over the "mode" attribute, with "Show In OSRA Project Explorer" highlighted in green.

The **OSRA Context Explorer** view at the bottom shows the properties for the selected "Interface Attribute mode". A red box highlights this view, which includes fields for Name, Data Type, Kind, Attribute Access Level, and Accessibility Level.

A successful project

- Obeo and ESA team satisfied by the activity
 - The project was done in time
 - Thanks to a good and close cooperation between both partners in the project
- An agile methodology really adapted
 - Allowing a quick project start up
 - And a progressive ramp up of the ESA team on the modeling technologies
- A small extension to the project will be soon finalized

