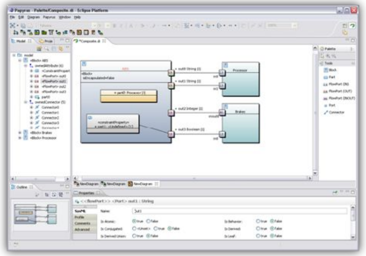
The evolution of Papyrus
An open UML and SysML tool

Camille Letavernier and Philip Langer
{cletavernier|planger}@eclipsesource.com

Papyrus

- Open Source UML modeling platform License EPL 2.0
  - Based on the Eclipse Modeling Framework
  - Based on modeling standards: UML, SysML, fUML, Alf, OCL, …
  - Supported by an active open-source community
  - Enables to build domain-specific tools based on UML, SysML, etc.

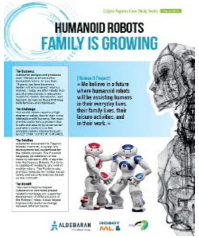


Ericsson Review

Design, architects, and complex communication systems: painting the bigger picture

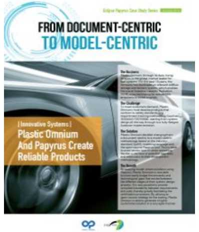
DESIGNING FREEDOM OF FLIGHT

« Papyrus allowed the SysML language to be customized to a form more readily understandable by Airbus System designers. »

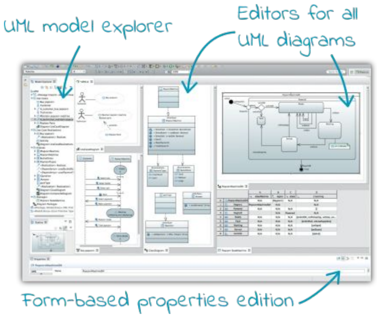

HUMANOID ROBOTS FAMILY IS GROWING

« We believe in a future where humanoid robots will be assisting humans in their everyday lives, their family lives, their leisure activities, and in their work. »

A COLLABORATION PLATFORM FOR AVIONICS SYSTEMS

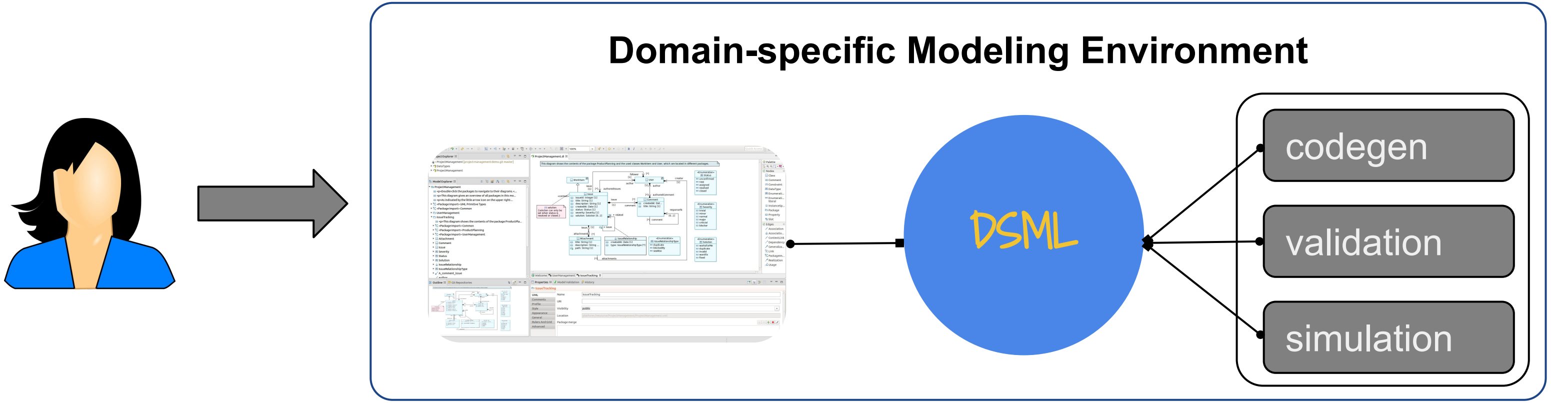

FROM DOCUMENT-CENTRIC TO MODEL-CENTRIC

Innovative Systems: Plastic Omnium And Papyrus Create Reliable Products



Domain-specific Modeling

- Model-based engineering is most successful if it is *domain-specific*
  - Highly customized modeling environments
  - Directly reflecting specific needs of a domain and its users
  - User roles, their backgrounds, methodologies, and tool chains


Domain-specific Modeling Environment

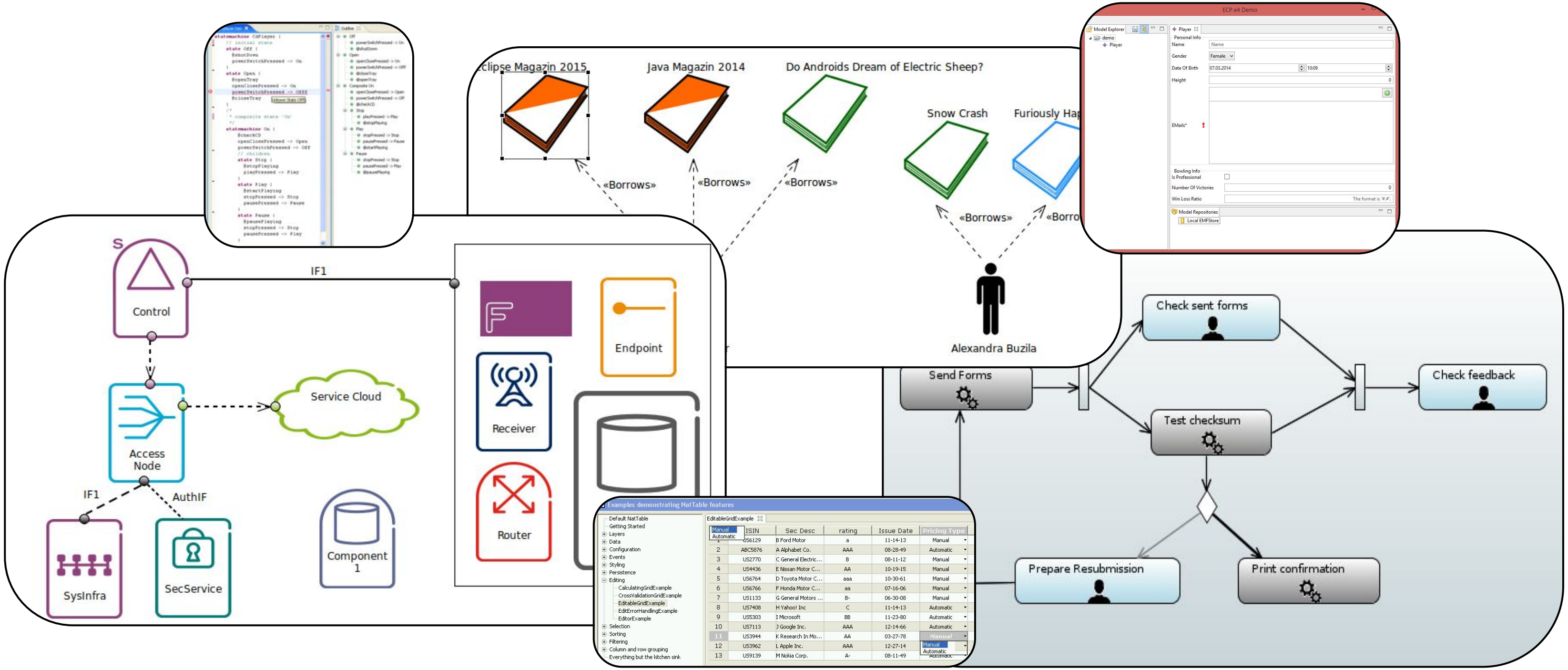
DSML

codegen

validation

simulation

Domain-specific Modeling



```
statemachine CdPlayer {
    // initial state
    state Off {
      @shutDown
      powerSwitchPressed -> On
    }
    state Open {
      @openTray
      openClosePressed -> On
      powerSwitchPressed -> Offf
      @closeTray
    }
    /*
     * composite state 'On'
     */
    statemachine On {
        @checkCD
        openClosePressed -> Open
        powerSwitchPressed -> Off
        // children
        state Stop {
          @stopPlaying
          playPressed -> Play
        }
        state Play {
          @startPlaying
          stopPressed -> Stop
          pausePressed -> Pause
        }
        state Pause {
          @pausePlaying
          stopPressed -> Stop
          pausePressed -> Play
        }
```

Unkown State Offf

Outline
- Off
  - powerSwitchPressed -> On
  - @shutDown
- Open
  - openClosePressed -> On
  - powerSwitchPressed -> Off
  - @closeTray
  - @openTray
- Composite On
  - openClosePressed -> Open
  - powerSwitchPressed -> Off
  - @checkCD
  - Stop
    - playPressed -> Play
    - @stopPlaying
  - Play
    - stopPressed -> Stop
    - pausePressed -> Pause
    - @startPlaying
  - Pause
    - stopPressed -> Stop
    - pausePressed -> Play
    - @pausePlaying

ECP e4 Demo

Model Explorer
- demo
  - Player

Player

Personal Info

Name: Name

Gender: Female

Date Of Birth: 07.03.2014 10:09

Height: 0

EMails* !

Bowling Info

Is Professional □

Number Of Victories: 0

Win Loss Ratio: The format is '#.#'.

Model Repositories
- Local EMFStore

Examples demonstrating NatTable features

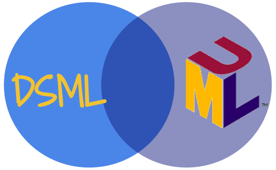
- Default NatTable
- Getting Started
- Layers
- Data
- Configuration
- Events
- Styling
- Persistence
- Editing
  - CalculatingGridExample
  - CrossValidationGridExample
  - EditableGridExample
  - EditErrorHandlingExample
  - EditorExample
- Selection
- Sorting
- Filtering
- Column and row grouping
- Everything but the kitchen sink

EditableGridExample

	ISIN	Sec Desc	rating	Issue Date	Pricing Type
1	US6129	B Ford Motor	a	11-14-13	Manual
2	ABC5876	A Alphabet Co.	AAA	08-28-49	Automatic
3	US2770	C General Electric...	B	08-11-12	Manual
4	US4436	E Nissan Motor C...	AA	10-19-15	Manual
5	US6764	D Toyota Motor C...	aaa	10-30-61	Manual
6	US6766	F Honda Motor C...	aa	07-16-06	Manual
7	US1133	G General Motors ...	B-	06-30-08	Manual
8	US7408	H Yahoo! Inc	C	11-14-13	Automatic
9	US5303	I Microsoft	BB	11-23-80	Automatic
10	US7113	J Google Inc.	AAA	12-14-66	Automatic
11	US3944	K Research In Mo...	AA	03-27-78	Manual
12	US3962	L Apple Inc.	AAA	12-27-14	
13	US9139	M Nokia Corp.	A-	08-11-49	Automatic

Domain-specific Modeling vs UML, SysML, etc.

- Standardized Modeling Languages (UML, SysML, …)
    - Reuse well-known and -proven language concepts
    - Reuse existing tools and components
    - Interoperability and connectability with other models
    - Conformance to industry standards

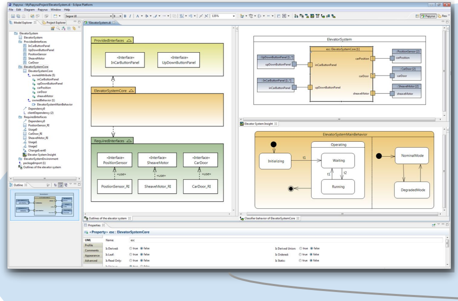


- Domain-specific Modeling vs. UML/SysML?
    - Contradiction? NO!
    - Papyrus can be used as a platform
        - Graphical syntax, palette, property views, editing behavior, etc.
        - Based on EMF and Eclipse RCP

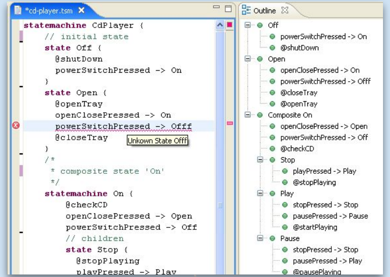
Domain-specific Modeling **with** UML, SysML, etc.

- Thanks to the great Open Source Eclipse Modeling Ecosystem...

Papyrus



Xtext



*cd-player.tsm

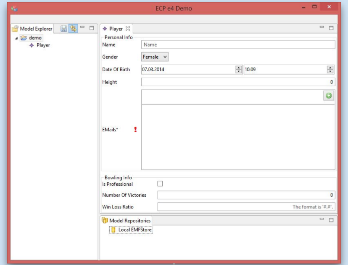
```
statemachine CdPlayer {
    // initial state
    state Off {
      @shutDown
      powerSwitchPressed -> On
    }
    state Open {
      @openTray
      openClosePressed -> On
      powerSwitchPressed -> Offf
      @closeTray
    }
    /*
     * composite state 'On'
     */
    statemachine On {
        @checkCD
        openClosePressed -> Open
        powerSwitchPressed -> Off
        // children
        state Stop {
          @stopPlaying
          playPressed -> Play
```

Unkown State Offf

Outline

```
Off
    powerSwitchPressed -> On
    @shutDown
Open
    openClosePressed -> On
    powerSwitchPressed -> Offf
    @closeTray
    @openTray
Composite On
    openClosePressed -> Open
    powerSwitchPressed -> Off
    @checkCD
    Stop
        playPressed -> Play
        @stopPlaying
    Play
        stopPressed -> Stop
        pausePressed -> Pause
        @startPlaying
    Pause
        stopPressed -> Stop
        pausePressed -> Play
        @pausePlaying
```

EMF Forms



ECP e4 Demo

Model Explorer: demo / Player

Player

Personal Info

Name: Name

Gender: Female

Date Of Birth: 07.03.2014 10:09

Height: 0

EMails*

Bowling Info

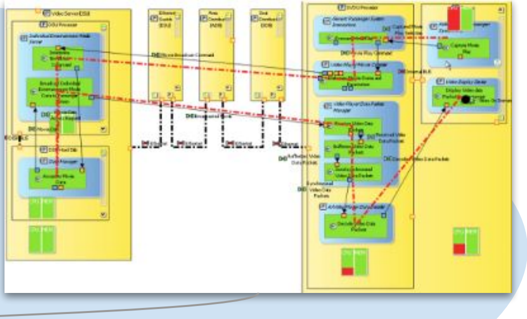
Is Professional ☐

Number Of Victories: 0

Win Loss Ratio: The format is '#.#'.

Model Repositories: Local EMFStore

Sirius



...

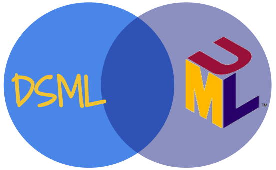
emf

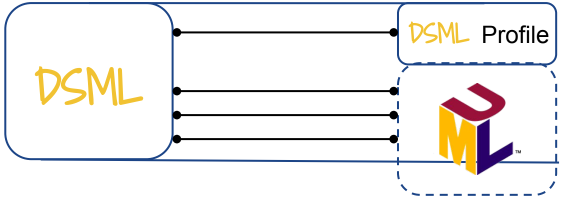
ECLIPSE MODELING FRAMEWORK

...

Papyrus as a Platform

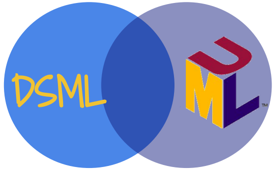
- Adding “domain-specificness” to UML
  - “Plain” UML is a general purpose modeling language
  - UML Profiles allow to *extend* UML with domain-specific concepts

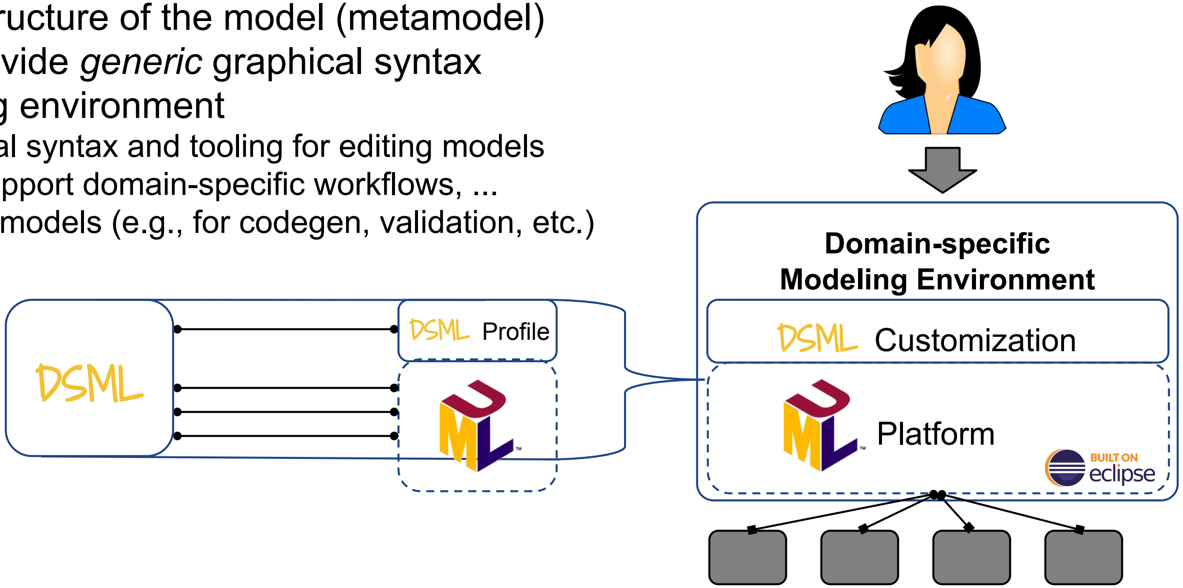




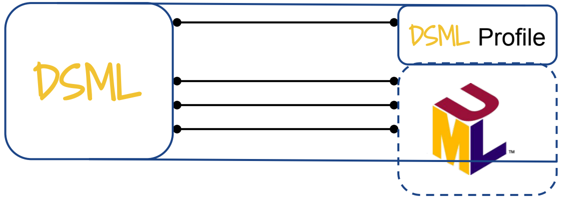
Papyrus as a Platform

- Building a domain-specific modeling tool
  - UML profile defines only structure of the model (metamodel)
  - Off-the-shelf UML tools provide *generic* graphical syntax
  - A domain-specific modeling environment
    - Domain-specific graphical syntax and tooling for editing models
    - Rich client platform to support domain-specific workflows, ...
    - Powerful API to process models (e.g., for codegen, validation, etc.)

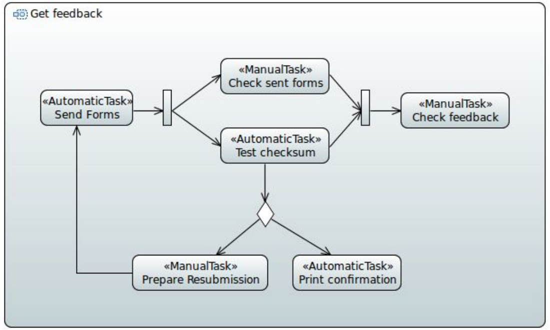


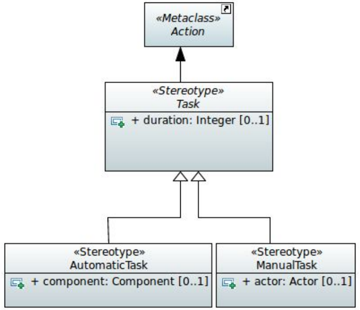


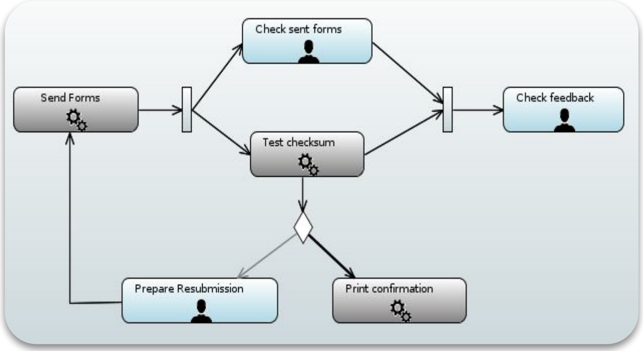
Example and Demo



- User Workflow Modeling Language
  - A workflow contains tasks
  - Manual task and automatic tasks
  - Flow between tasks and decisions
  - Duration, responsibilities, and probabilities

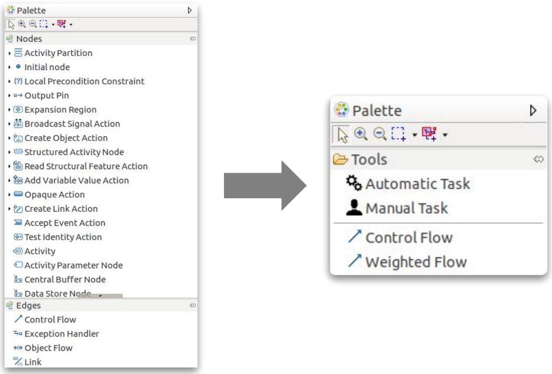

Get feedback
«AutomaticTask» Send Forms
«ManualTask» Check sent forms
«AutomaticTask» Test checksum
«ManualTask» Check feedback
«ManualTask» Prepare Resubmission
«AutomaticTask» Print confirmation

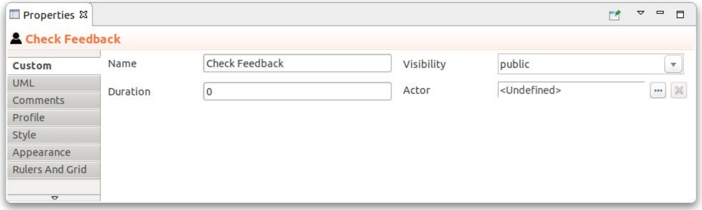

«Metaclass» Action
«Stereotype» Task
+ duration: Integer [0..1]
«Stereotype» AutomaticTask
+ component: Component [0..1]
«Stereotype» ManualTask
+ actor: Actor [0..1]



Custom Tooling

- Abstract and concrete syntax is just the beginning of a custom tool
  - More customizations are required for a domain-specific modeling environment
  - Papyrus supports full customizability of palette, property views, menus, ...
  - Papyrus builds on Eclipse → rich tool development platform
    - Integration with other tools, code generators, etc.
    - Wizards, menus, toolbars, views, ...





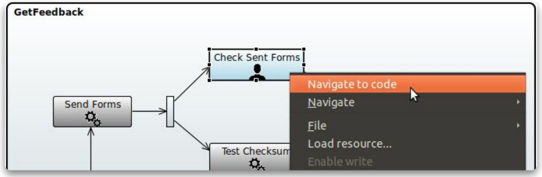
Check Feedback

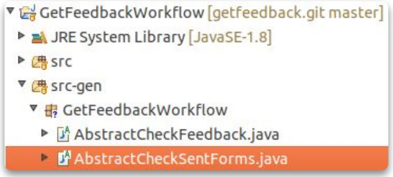
Name: Check Feedback

Visibility: public

Duration: 0

Actor: <Undefined>

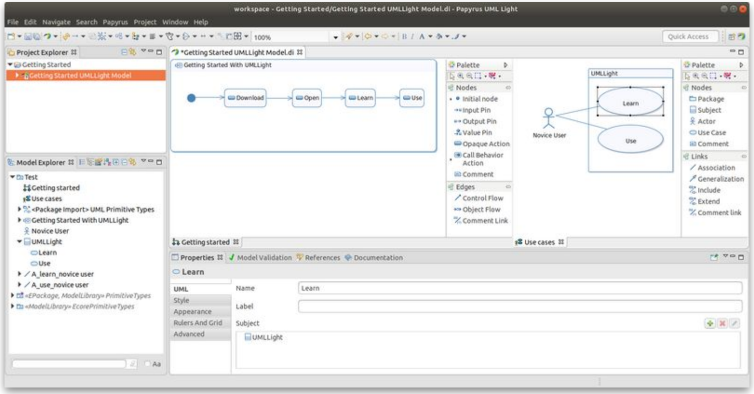



GetFeedbackWorkflow [getfeedback.git master]
JRE System Library [JavaSE-1.8]
src
src-gen
GetFeedbackWorkflow
AbstractCheckFeedback.java
AbstractCheckSentForms.java

What’s new around Papyrus?

- Papyrus UMLLight
  - Optimization of Papyrus for the most common UML features
  - Simplified menus, wizards, palettes, property views, fresh diagram style, etc.
  - Source code and customization guide publicly available *
  - Funded by the Papyrus Industry Consortium
  - Implemented by EclipseSource

- SysML 1.6
  - Implementation of the profile
  - Element and diagram types
  - Migration tool from SysML 1.4 to 1.6
  - Implemented by CEA



Name: Learn

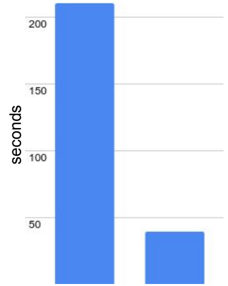
Label:

Subject: UMLLight

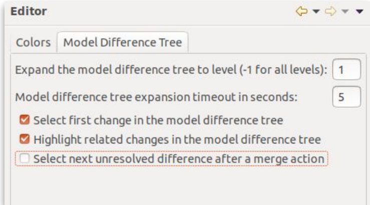
* github.com/eclipsesource/papyrus-umllight

What’s new around Papyrus?

- Papyrus Compare
  - Allows diffing and merging Papyrus models
  - Enables managing Papyrus models in Git repository
  - Enhancements of the last year
    - Huge performance and usability improvements
    - Changing merge and conflict resolution decisions
    - Property compare



Git Merge
~ 250 diagrams
~ 230.000 diagram elements
~ 55.000 semantic elements



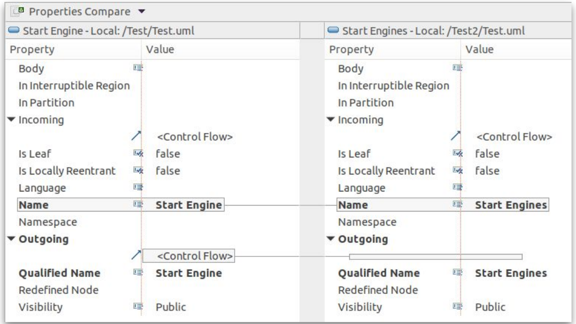
Editor

Colors | Model Difference Tree

Expand the model difference tree to level (-1 for all levels): 1

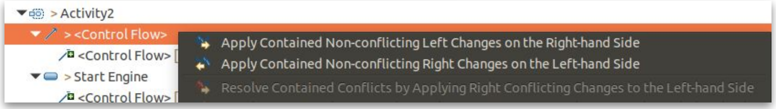
Model difference tree expansion timeout in seconds: 5

☑ Select first change in the model difference tree
☑ Highlight related changes in the model difference tree
□ Select next unresolved difference after a merge action



Properties Compare

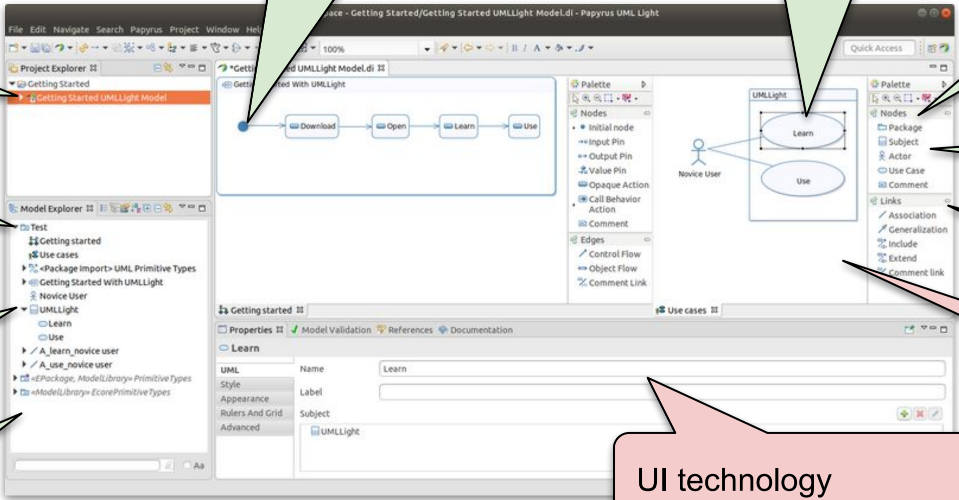
Start Engine - Local: /Test/Test.uml		Start Engines - Local: /Test2/Test.uml	
Property	**Value**	**Property**	**Value**
Body		Body	
In Interruptible Region		In Interruptible Region	
In Partition		In Partition	
Incoming		Incoming	
	<Control Flow>		<Control Flow>
Is Leaf	false	Is Leaf	false
Is Locally Reentrant	false	Is Locally Reentrant	false
Language		Language	
Name	**Start Engine**	**Name**	**Start Engines**
Namespace		Namespace	
Outgoing		**Outgoing**	
	<Control Flow>		
Qualified Name	**Start Engine**	**Qualified Name**	**Start Engines**
Redefined Node		Redefined Node	
Visibility	Public	Visibility	Public


> Activity2
> <Control Flow>
<Control Flow>
> Start Engine
<Control Flow>
Apply Contained Non-conflicting Left Changes on the Right-hand Side
Apply Contained Non-conflicting Right Changes on the Left-hand Side
Resolve Contained Conflicts by Applying Right Conflicting Changes to the Left-hand Side

wiki.eclipse.org/Papyrus_Compare

What’s next with Papyrus?

- Today and tomorrow



Rendering definition

Editing commands

Palette definition

Serialization

Element types

Model management

Customizability

Language types

Rendering technology

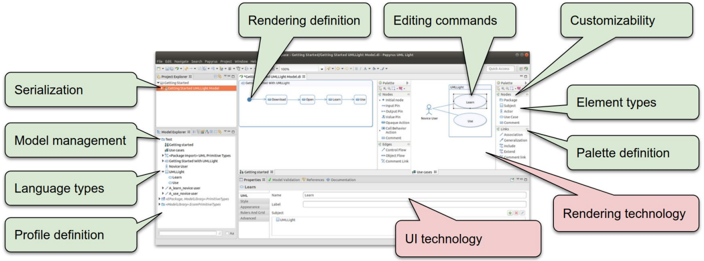
Profile definition

UI technology

| http://eclipsesource.com | Camille Letavernier and Philip Langer | The evolution of Papyrus, an open SysML and UML tool

What’s next with Papyrus?

- Aging UI & Rendering Technology
  - Papyrus depends on GMF
  - GMF depends on GEF3
  - Rich-client: SWT → JavaFX
  - Browser-based: SWT → HTML5/CSS/SVG
- Lots of functionality
  - Hardly possible and desirable to re-implement
- How can we keep a lot of the functionality?
  - But evolve the UI & rendering technology



Rendering definition

Editing commands

Customizability

Serialization

Element types

Model management

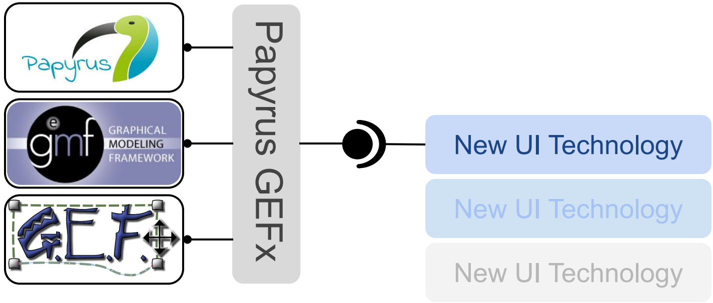
Palette definition

Language types

Rendering technology

Profile definition

UI technology



Papyrus

GRAPHICAL MODELING FRAMEWORK

G.E.F.

Papyrus GEFx

New UI Technology

New UI Technology

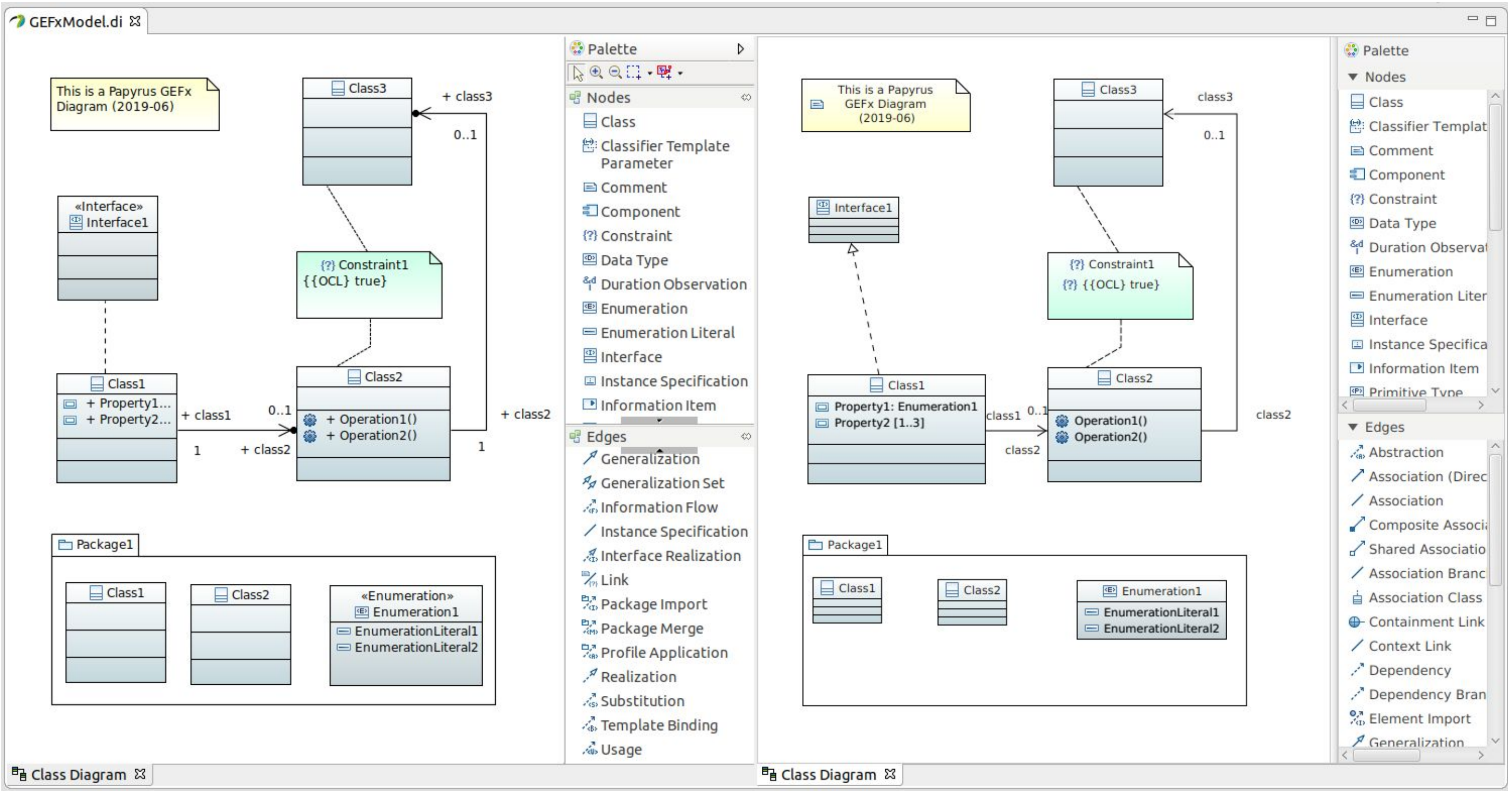
New UI Technology

Papyrus GEFx: Papyrus with GEF5.x

- Goals
  - Migration to **modern** diagram components & UI
  - More **modular** architecture, improving extension & customization
  - Compatibility has to be preserved: **incremental** migration
- How
  - Remove the GEF 3.x (“Draw2D”) parts of Papyrus Diagram UI
  - Re-implement Papyrus Diagram UI with GEF 5.x (“JavaFX”)
- Result
  - Papyrus Diagrams based on GEF 5.x
  - Compatible with the existing Papyrus Editor & Models

Papyrus GEFx

- GEFx Connector
  - Some generic services & diagram parts on top of GEF (5.x, JavaFX)
- GMF Connector
  - Implement GEFx services (Notation, EMF Transactions)
  - Base interactions (Move, Resize, Create, etc.)
- Papyrus Integration
  - Semantic models
  - Customization models & CSS (Edition logic, Palettes, Diagram Structure, ...)
  - Retain editor integration with Model Explorer & Properties


GEFxModel.di
This is a Papyrus GEFx Diagram (2019-06)
Class3
+ class3
0..1
«Interface»
Interface1
{?} Constraint1
{{OCL} true}
Class1
+ Property1...
+ Property2...
+ class1
0..1
1
+ class2
Class2
+ Operation1()
+ Operation2()
+ class2
1
Package1
Class1
Class2
«Enumeration»
Enumeration1
EnumerationLiteral1
EnumerationLiteral2
Class Diagram
Palette
Nodes
Class
Classifier Template Parameter
Comment
Component
Constraint
Data Type
Duration Observation
Enumeration
Enumeration Literal
Interface
Instance Specification
Information Item
Edges
Generalization
Generalization Set
Information Flow
Instance Specification
Interface Realization
Link
Package Import
Package Merge
Profile Application
Realization
Substitution
Template Binding
Usage
This is a Papyrus GEFx Diagram (2019-06)
Class3
class3
0..1
Interface1
{?} Constraint1
{?} {{OCL} true}
Class1
Property1: Enumeration1
Property2 [1..3]
class1
0..1
class2
Class2
Operation1()
Operation2()
class2
Package1
Class1
Class2
Enumeration1
EnumerationLiteral1
EnumerationLiteral2
Class Diagram
Palette
Nodes
Class
Classifier Templat
Comment
Component
Constraint
Data Type
Duration Observat
Enumeration
Enumeration Liter
Interface
Instance Specifica
Information Item
Primitive Type
Edges
Abstraction
Association (Direc
Association
Composite Associ
Shared Associatio
Association Branc
Association Class
Containment Link
Context Link
Dependency
Dependency Bran
Element Import
Generalization

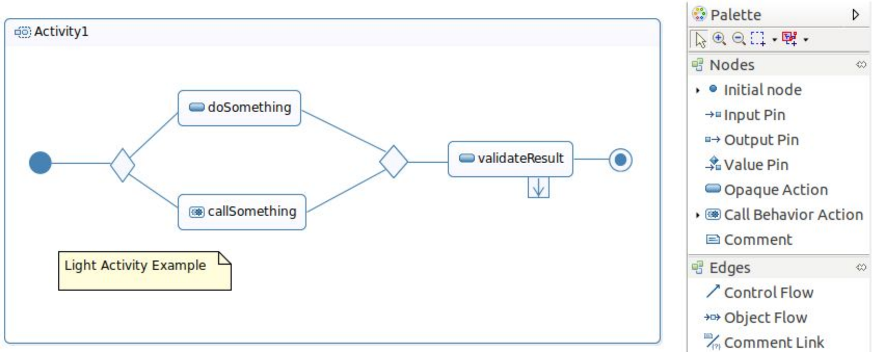
Papyrus GEFx

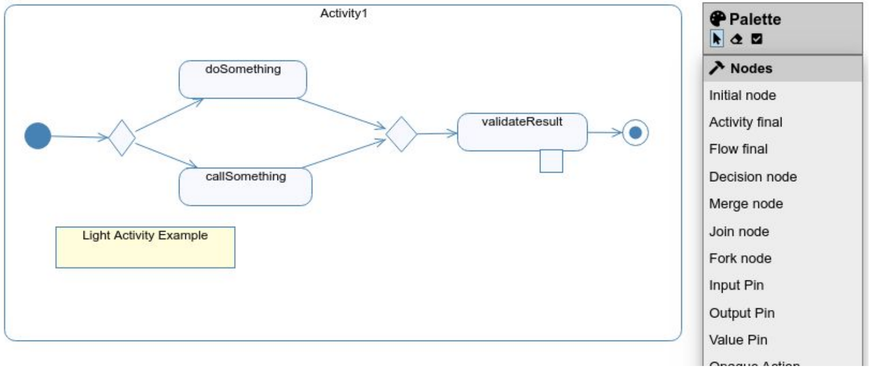
- Integration effort
  - Custom figures & labels
  - Custom user interaction
  - Custom ... (depends on the diagram implementation)
- Benefits
  - Modern UI toolkit (JavaFX)
  - Compatibility with Papyrus, including customizations
  - Reuse of GMF Runtime (or EMF/UML)
  - Integrated with Eclipse, but not based on it
  - Improved architecture, leveraging services & dependency-injection

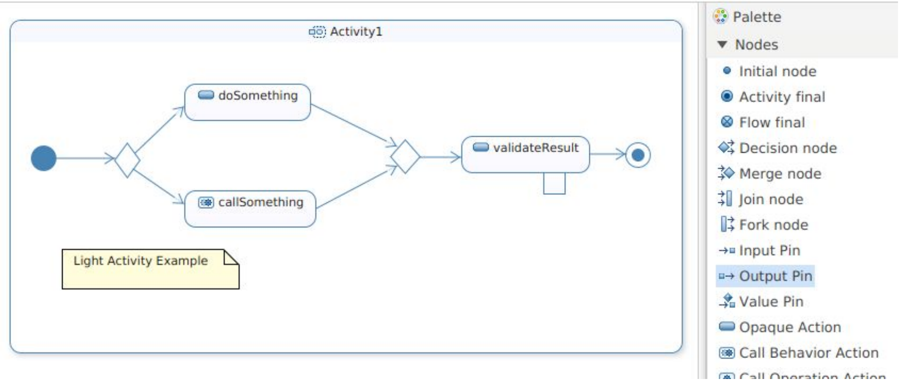
Papyrus GEFx with GLSP: Web-based Papyrus

Diagrams in web and space with GLSP
Wednesday, 15:10 - 15:45, Bürgersaal 2

- GLSP (Graphical Language Server Protocol)
  - Enables browser-based clients
  - Protocol for abstracting the diagram logic from the rendering technology
- GLSP connector
  - Server side: based on Papyrus-GEFx
  - Client side: generic web client based on GLSP client (Theia, Sprotty)
- Any diagram supported by Papyrus-GEFx can be opened with GLSP-GEFx
  - Structure & edition is supported by the server
  - Server can control styling (via CSS classes and diagram structure)
  - But implementation is required on the client for rendering







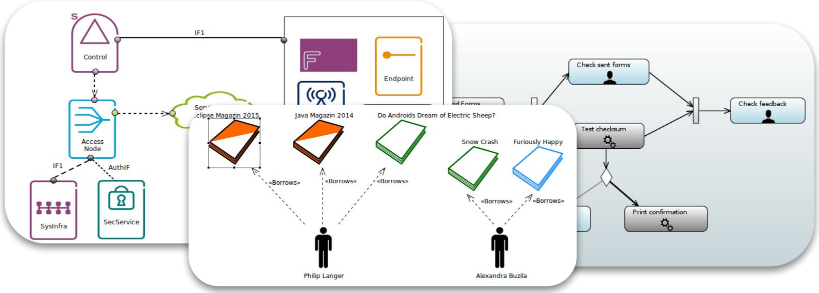
Top Left: Papyrus (GMF + Draw2D)

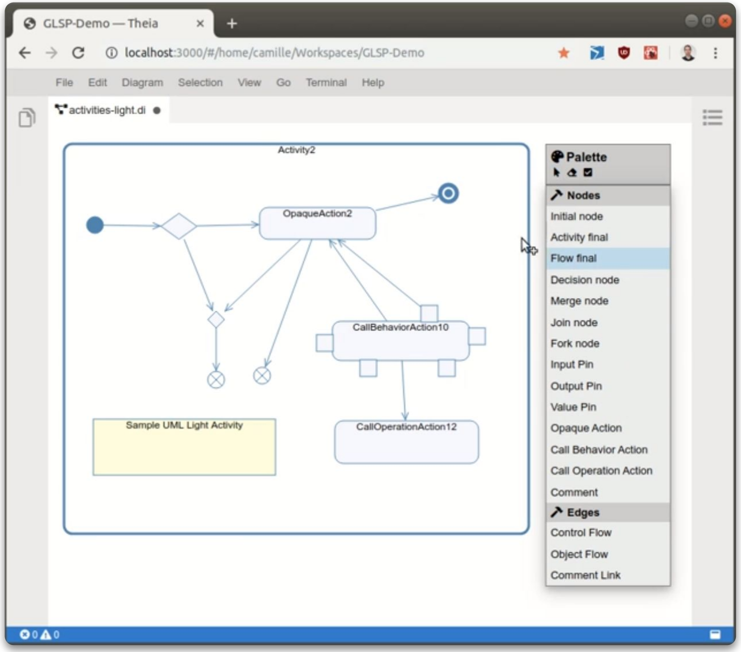
Bottom Left: Papyrus-GEFx (JavaFX)

Top Right: GEFx-GLSP (Web-based, Sprotty/Theia)

Thanks a lot!

- Contact us for more details
  - planger@eclipsesource.com
  - cletavernier@eclipsesource.com
- Links
  - https://www.eclipse.org/papyrus/download.html
  - https://github.com/eclipsesource/papyrus-umllight
  - https://wiki.eclipse.org/Papyrus_Compare
  - https://github.com/eclipsesource/papyrus-gefx
  - https://github.com/eclipsesource/papyrus-gefx-glsp


Control
IF1
Endpoint
Access Node
IF1
AuthIF
SysInfra
SecService
Eclipse Magazin 2015
Java Magazin 2014
Do Androids Dream of Electric Sheep?
Snow Crash
Furiously Happy
«Borrows»
«Borrows»
«Borrows»
«Borrows»
«Borrows»
Philip Langer
Alexandra Buzila
Check sent forms
Test checksum
Check feedback
Print confirmation


GLSP-Demo — Theia
localhost:3000/#/home/camille/Workspaces/GLSP-Demo
File
Edit
Diagram
Selection
View
Go
Terminal
Help
activities-light.di
Activity2
OpaqueAction2
CallBehaviorAction10
CallOperationAction12
Sample UML Light Activity
Palette
Nodes
Initial node
Activity final
Flow final
Decision node
Merge node
Join node
Fork node
Input Pin
Output Pin
Value Pin
Opaque Action
Call Behavior Action
Call Operation Action
Comment
Edges
Control Flow
Object Flow
Comment Link