

Papyrus Information Modeling

Demo of Pre-release Beta Build

Philip Langer and Benoit Maggi

planger@eclipsesource.com || benoit.maggi@cea.fr

Agenda

- Context
- Work done so far
- Demo of the current state
- Next steps
- Your feedback

Context

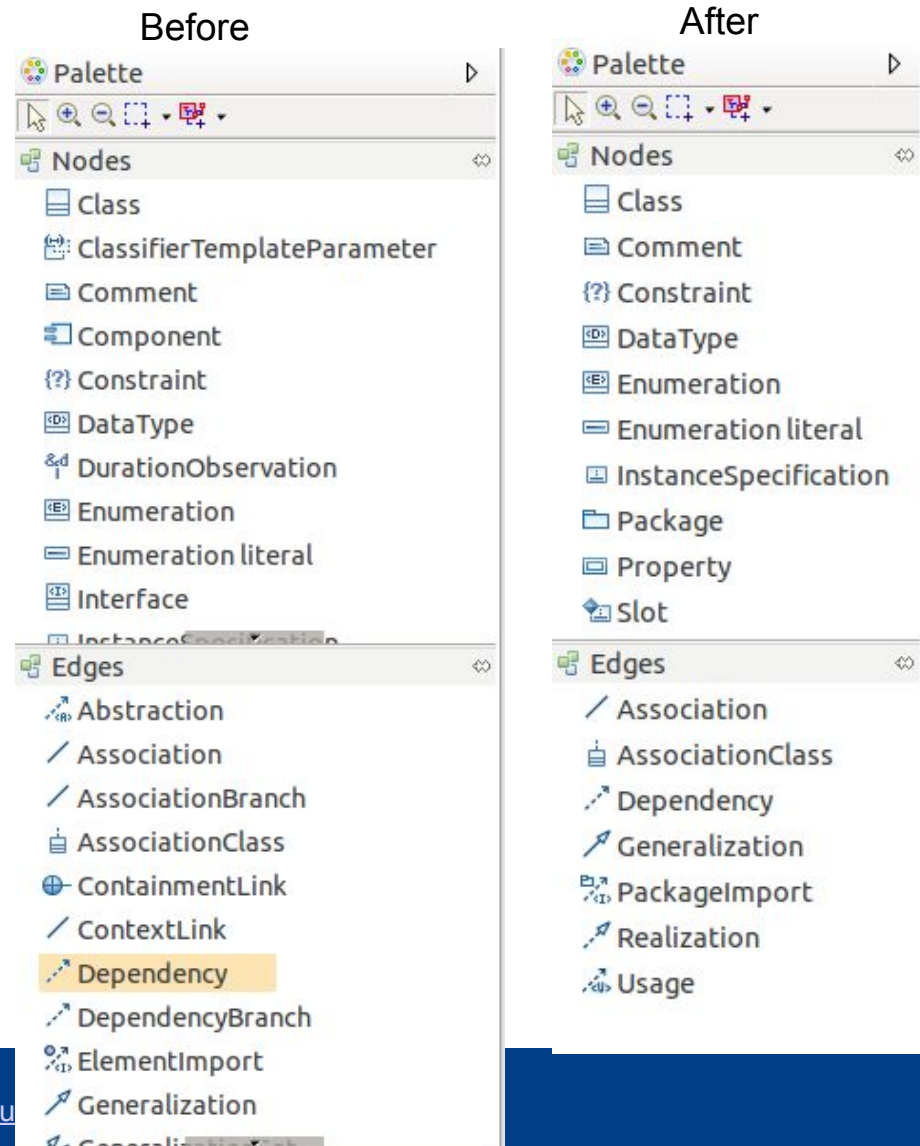
- **Objective: Information Modeling product**
 - Based on the Papyrus platform
 - Streamlined for information modeling with UML class diagrams
 - Targeted at the ease of use for users new to Papyrus
 - Reduce the complexity of the Eclipse UI
 - Illustrate the customizability of Papyrus and the Eclipse platform
- **Project details**
 - Carried out by CEA and EclipseSource
 - Project size is 36 developer days ($\sim \frac{2}{3}$ spent)
 - Project start was mid May
 - Release planned for end of June (targeted platform: Neon)
- **Funded by the Papyrus Industrial Consortium**
<https://www.polarsys.org/ic/papyrus>

Work done so far

- Identification of the UML subset used for Information Modeling
 - Based on sample models we retrieved
- Implementation of the UML subset as a Papyrus viewpoint
 - Simplification of the Papyrus UI
 - Palette, property views, etc.
- Implementation of the simplification of the Eclipse UI
 - Toolbars, wizards, menu actions, etc.
- Implementation of a diagram style
 - Optimized for information modeling
- Documentation
 - Eclipse Wiki
 - Importable example model
 - Getting started tutorial based on Eclipse Cheat Sheets
- Technical tasks excluded in the list above
 - Setup of continuous integration build and development infrastructure, product configuration and target platform specification, separation of composable and non-composable plug-ins, code reviews for best practices, ...

Work done so far

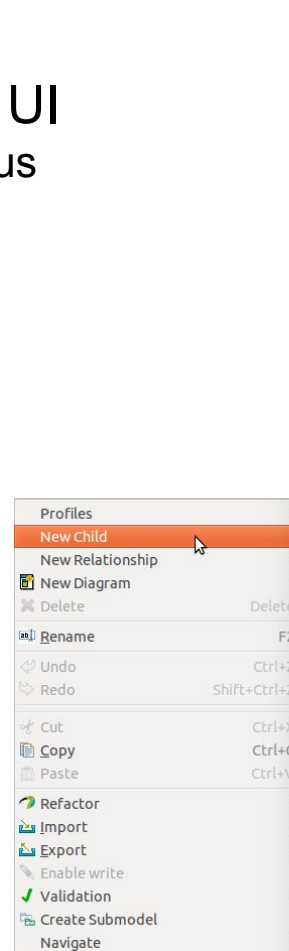
- Simplification: Papyrus UI
 - Palette



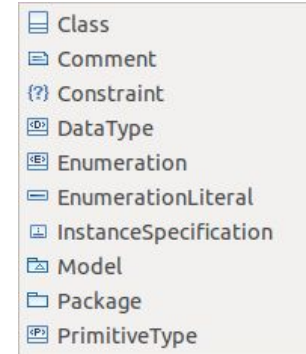
Work done so far

- Simplification: Papyrus UI
 - New child creation menus

Before

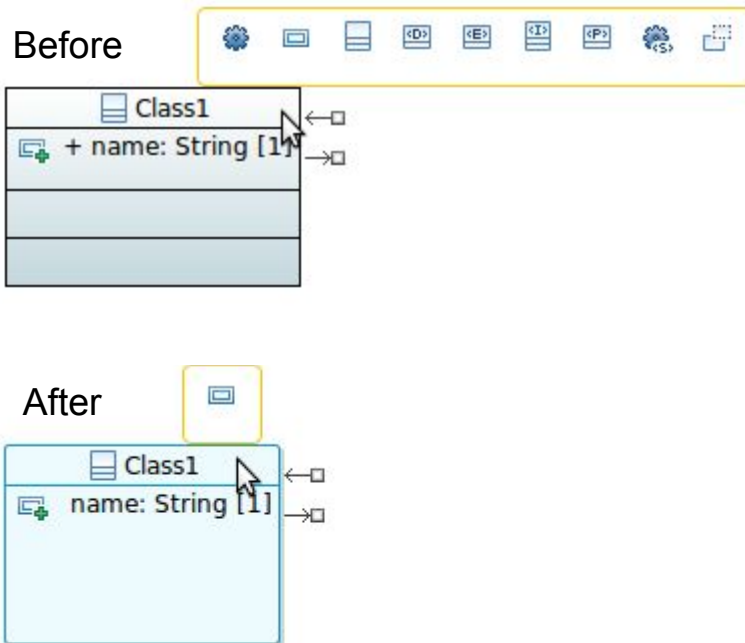


After



Work done so far

- Simplification: Papyrus UI
 - Modeling assistants and diagram style



Work done so far

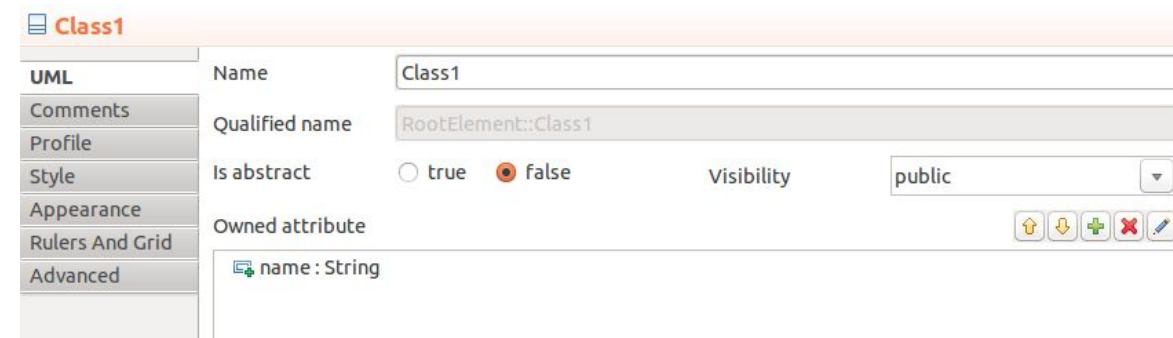
- Simplification: Papyrus UI
 - Property views

Before



The 'Before' screenshot shows the Papyrus UI property view for a class named 'Class1'. The 'UML' tab is selected. The 'Name' field contains 'Class1'. The 'Qualified name' field contains 'RootElement::Class1'. The 'Is abstract' field has radio buttons for 'true' and 'false', with 'false' selected. The 'Is active' field has radio buttons for 'true' and 'false', with 'false' selected. The 'Is leaf' field has radio buttons for 'true' and 'false', with 'false' selected. The 'Visibility' field is a dropdown menu set to 'public'.

After

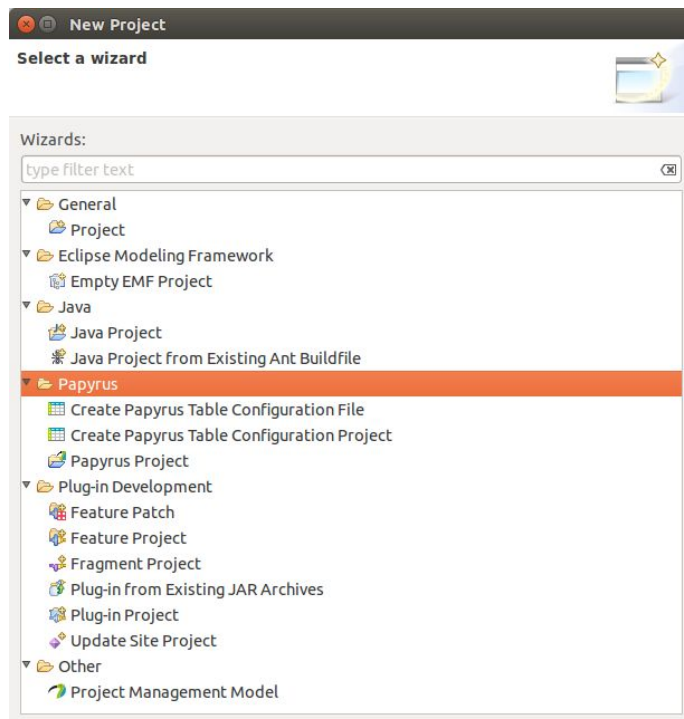


The 'After' screenshot shows the Papyrus UI property view for 'Class1' after simplification. The 'UML' tab is selected. The 'Name' field contains 'Class1'. The 'Qualified name' field contains 'RootElement::Class1'. The 'Is abstract' field has radio buttons for 'true' and 'false', with 'false' selected. The 'Visibility' field is a dropdown menu set to 'public'. A new 'Owned attribute' section has been added, containing a single attribute: 'name : String'. The attribute list has icons for adding, deleting, and editing.

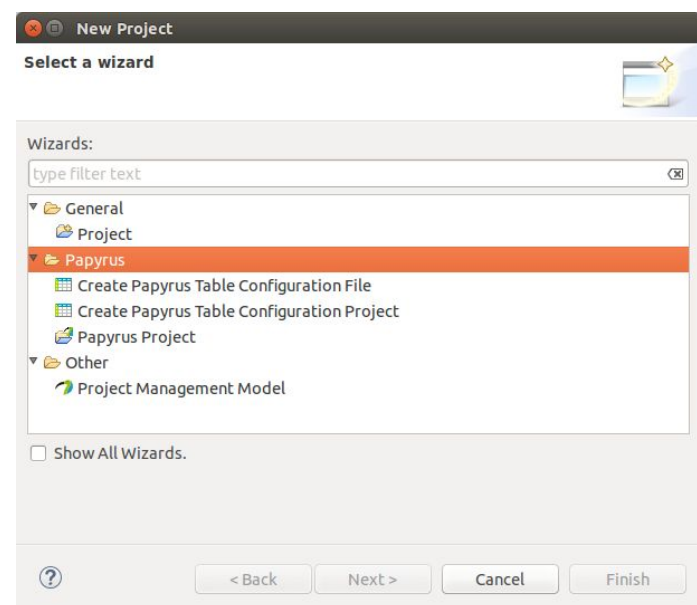
Work done so far

- Simplification: Eclipse UI
 - Wizards

Before



After

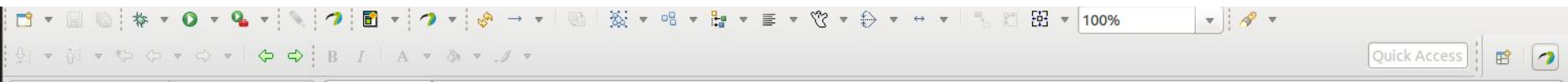


Work done so far

- Simplification: Eclipse UI

- Toolbars

Before

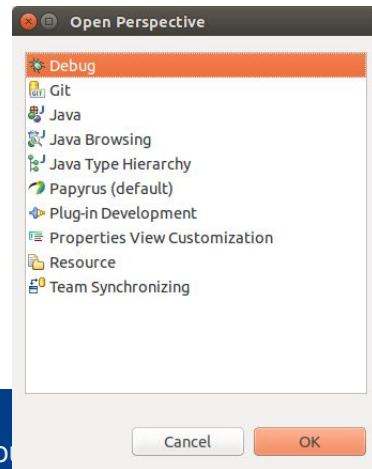


After

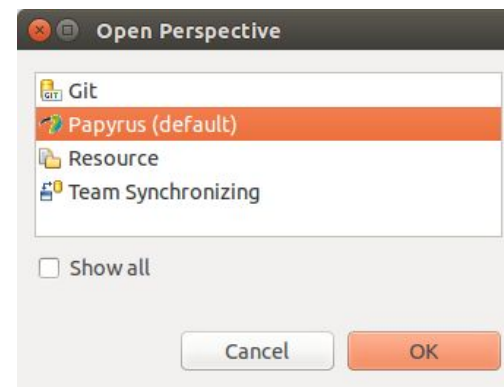


- Perspectives

Before



After



workspace-papyrus - Papyrus - ProjectManagement/ProjectManagement.di - Papyrus Information Modeling

Project Explorer

- ProjectManagement
 - DataTypes
 - ProjectManagement

Model Explorer

- ProjectManagement
 - DataTypes
 - «EPackage, ModelLibrary» UML Prim
 - «ModelLibrary» Ecore Primitive Type

Outline

ProjectManagement.di

This diagram shows the contents of the package ProductPlanning and the used classes WorkItem and User, which are located in different packages.

{??} solution {solution can only be set when status is resolved or closed.}

```

classDiagram
    class WorkItem
    class Issue {
        +issueId: Integer [1]
        +title: String [1]
        +description: String [1]
        +createdAt: Date [1]
        +status: Status [1]
        +severity: Severity [1]
        +solution: Solution [0..1]
    }
    class Attachment {
        +title: String [1]
        +description: String...
        +path: EString [1]
    }
    class IssueRelationship {
        +createdAt: Date [1]
        +type: IssueRelationshipType [*]
    }
    class User {
        +author: User [1]
    }
    class Comment {
        +createdAt: Date [1]
        +title: String [1]
    }
    class Status {
        +unconfirmed
        +new
        +assigned
        +resolved
        +closed
    }
    class Severity {
        +trivial
        +minor
        +normal
        +major
        +critical
        +blocker
    }
    class Solution {
        +worksForMe
        +duplicate
        +blockedBy
        +invalid
        +wontFix
        +fixed
    }
    class IssueRelationshipType {
        +duplicate
        +blockedBy
        +seeAlso
    }

    WorkItem <|-- Issue
    Issue "1" -- "*" Attachment : attachments
    Issue "1" -- "*" IssueRelationship : related
    Issue "1" -- "*" User : author
    User "1" -- "*" User : follower
    User "1" -- "1" User : creator
    Issue "1" -- "*" Comment : authoredIssues
    Comment "1" -- "*" User : author
    Comment "1" -- "*" Comment : authoredComment
    Comment "1" -- "0..1" Comment : responseTo
    Issue "1" -- "*" Issue : comment
    Issue "1" -- "*" Issue : issue
    
```

Properties

Model Validation

Issue

UML Name: Issue

Qualified name: ProjectManagement::IssueTracking::Issue

Profile:

Style: Is abstract: true false

Appearance: Visibility: public

Owned attribute:

- issueId : Integer
- title : String
- description : String

Next steps

- Finalize documentation and user guidance
 - Getting started tutorial as a browser version
 - Coordination with work on general Papyrus docs
 - Dissemination on the Papyrus web page
- Address most important open issues
 - Filter types to UML subsets when creating types from property view
 - Missing Open / Open with menu entries for non-Papyrus files
 - ... (see [bugzilla](#))
- Get your feedback on requirements and enhancements
 - Create a backlog for future development
 - Find partners to support us in the future development

All contributions more than welcome (reqs, testing, funding, etc.)

Feedback

- General feedback
- Diagram types
 - Is the identified subset adequate?
 - Which other diagram types would you would like to see?
- UML profiles
 - Standard UML profiles in the information modeling domain?
 - Implement easier way to create and share custom profiles?
- Import and export
 - From and to other tools?
 - Export of documentation?
- Code generation
 - Are there standard code generators you would like to have?
 - Implement easier way to create and share custom code generators?

Invitation to Contribute License EPL 1.0

- Get Papyrus for Information Modeling
 - Developer builds (see [wikipage](#))
 - Release coming shortly after Neon release of Eclipse
- Feedback, requirements, and bugs
 - Contact us (planger@eclipsesource.com; benoit.maggi@cea.fr)
 - Raise bug reports ([bugzilla](#))
 - Papyrus mailing list (mdt-papyrus.dev)
 - ...
- Fixes and enhancements
 - Push for review via [gerrit](#)
 - See also contributor guidelines (view [wikipage](#))
- Funding
 - Sponsored OS development (maintenance and your features)
 - Custom enhancements on top of Papyrus for Information Modeling