Papyrus Industry Consortium
Steering Committee

Subject: Request for Work: Papyrus UMLLight[1] Development

Eclipse Papyrus is not only an open-source UML modeling tool, but also a platform for creating domain- and methodology-specific UML-based modeling tools. The Papyrus UML modeling tool, Papyrus UML for short, is not streamlined for a particular methodology, user group, or domain and, hence, is rather a pure implementation of the UML metamodel, exposing all its concepts and features without particular guidance. The generic and pure implementation of the UML metamodel in Papyrus UML is on purpose: Papyrus UML is not meant to be optimized for a particular user group or application domain; it shall rather act as basis for users creating domain- or methodology-specific, streamlined modeling tool based on the Papyrus platform.

However, many interested, novice users begin to use Papyrus as a plain UML modeling tool rather than as a platform. As Papyrus UML is not simplified and streamlined for novice users, such users can be easily get scared off by the number of menu items and commands representing UML concepts and features and, as a result, may lose interest in Papyrus before even getting to know the potential of the Papyrus platform and its ecosystem.

In order to nurture this important channel to attract new users, a simplified UML modeling tool is required. Such a simplified UML modeling tool would reduce the number of UML concepts that can be used in Papyrus to the most common UML subset. Such a simplified UML tool could then be recommended to new users and also be used in teaching and trainings.

The goal of this work is, therefore, to build a simplified UML modeling tool, called Papyrus UML Light. Papyrus UMLLight shall only expose the most common concepts and diagram types of UML (see 1.2) and guide the user in applying these concepts and diagram types.

This Request for Work is intended to serve as a basis for the supplier to deliver Statement of Works to the Papyrus IC Steering Committee. The Steering Committee members will then decide to launch the activity based on the Statement of Work they will receive.

1. REQUESTED WORK

1.1. PAPYRUS UMLLIGHT

Papyrus UMLLight is a standalone modeling tool that is to be built on the Papyrus platform. Therefore, Papyrus UMLLight provides a simplified and streamlined modeling tools for the most common concepts of UML (see Section 1.2). Therefore, it shall customize the available palette tools, new child menus, property sheets, wizards for creating models, etc., in order to provide an optimized experience for novice users. Moreover, every feature and functionality that is not required by novice users shall be hidden to avoid confusion of novice users.

1.2. DIAGRAM TYPES AND UML SUBSET // TODO TO BE UPDATED

1. UML Use Case
    a. Package
    b. Subject
    c. Actor
    d. Use Case

[1] Final name to be approved

    e. Comment
    f. Association
    g. Generalization
    h. Include
    i. Extend
    j. Comment link

2. Activity Diagram
    a. Activity
    b. Initial Node
    c. Final Node
    d. Activity final
    e. Decision node
    f. Merge node
    g. Join node
    h. Fork node
    i. Opaque Action
    j. Comment
    k. Control Flow
    l. Link
    m. Action
    n. Call BehaviorAction
    o. CallOperation Action
    p. AcceptEventAction
    q. Pin
    r. InputPin
    s. OutputPin
3. Class Diagram
    a. Class
    b. Comment
    c. DataType
    d. Enumeration
    e. Enumeration Literal
    f. Interface
    g. Interface Realization
    h. Primitive Type
    i. Operation
    j. Package
    k. Parameter
    l. Aggregation (composition, aggregation)
    m. Association
    n. Association Class
    o. Generalization
    p. Link
    q. Realization
    r. Visibility (public, private, protected)
    s. Dependency
    t. Multiplicity
4. State Machine
    a. State
    b. Initial
    c. FinalState
    d. Choice
    e. Junction
    f. Comment
    g. Transition

    h. Link
5. Package Diagram
    a. Comment
    b. Model
    c. Package
    d. Containment Link
    e. Dependency
    f. Link
    g. PackageImport
6. Sequence Diagram*
    a. OccurrenceSpecification
    b. Executation Specification
    c. Interaction
    d. Action Execution Specification
    e. Comment
    f. Interaction Use
    g. Lifeline
    h. Message Async
    i. Message Sync
    j. Message Occurence Specification
    k. MessageSort
    l. MessageEnd
    m. CallEvent
    n. SignalEvent
    o. OpaqueBehavior
    p. Link

Also the property views for the UML elements listed above should be simplified and reduced. Which property to include/exclude will be decided together with the PapyrusIC during the project.

* Sequence Diagrams shall be included in Papyrus UMLLight, but this work is not intended to improve the Sequence Diagram editing support of Papyrus itself. On mutual agreement with the PapyrusIC, it may be decided to exclude Sequence Diagrams during the project, if the usability of Sequence Diagrams is decided to be not sufficient for our target audience.

1.3. Papyrus Photon Basis and Documentation of the Applied Customizations

Papyrus UMLLight shall be implemented based on the Papyrus version that is shipped with the simultaneous release of Eclipse Photon. This work must also include documenting the customizations applied during development allowing others to reproduce similar customizations. The documentation must be available publically, ideally in the Eclipse Wiki.

1.4. Expected Delivery and Sign-off

The deliverables of this project are

- Ready-to-use Eclipse-based product for Windows, Linux, and Mac
- Sources for building and maintaining this product, including the build configurations (sources must be published under the terms of the EPL v2.0)
- Automated maven-tycho-based build configuration
- Documentation of the customizations that have been applied (see Section 1.3)

A representative nominated by the PapyrusIC will be designated to monitor the progress and work closely together with the development team to provide detailed requirements and priorities for work items. This representative of the PapyrusIC will further evaluate the results at defined milestones, as well as at the project end to assess whether the requirements are fulfilled or not. Taking the overall development capacity of this contract into consideration and in coordination with the organization performing this work, the representative of the PapyrusIC will sign-off the deliverables once all requirements listed in this document have been addressed and no breaking issues are found that are not caused by Papyrus or other base technologies.

2. MODE OF THE WORK

Papyrus UMLLight development shall be performed in an open (visible in public) and agile manner, together with a representative of the PapyrusIC. Therefore, we plan to have biweekly phone calls to discuss the progress and define the next steps together with the supplier.

The supplier should also have access to Papyrus committers so that potentially required changes in the Papyrus platform can be applied for accommodating certain aspects in Papyrus UMLLight.