



Java Workflow Tooling (JWT)

Helios Review

Graduation/Release Review

Marc Dutoo (Open Wide, FR)

Christian Saad (University of Augsburg, DE)

Florian Lautenbacher (University of Augsburg, DE)



Overview



- About JWT
- Release 1.0
- Communication

About JWT - Overview



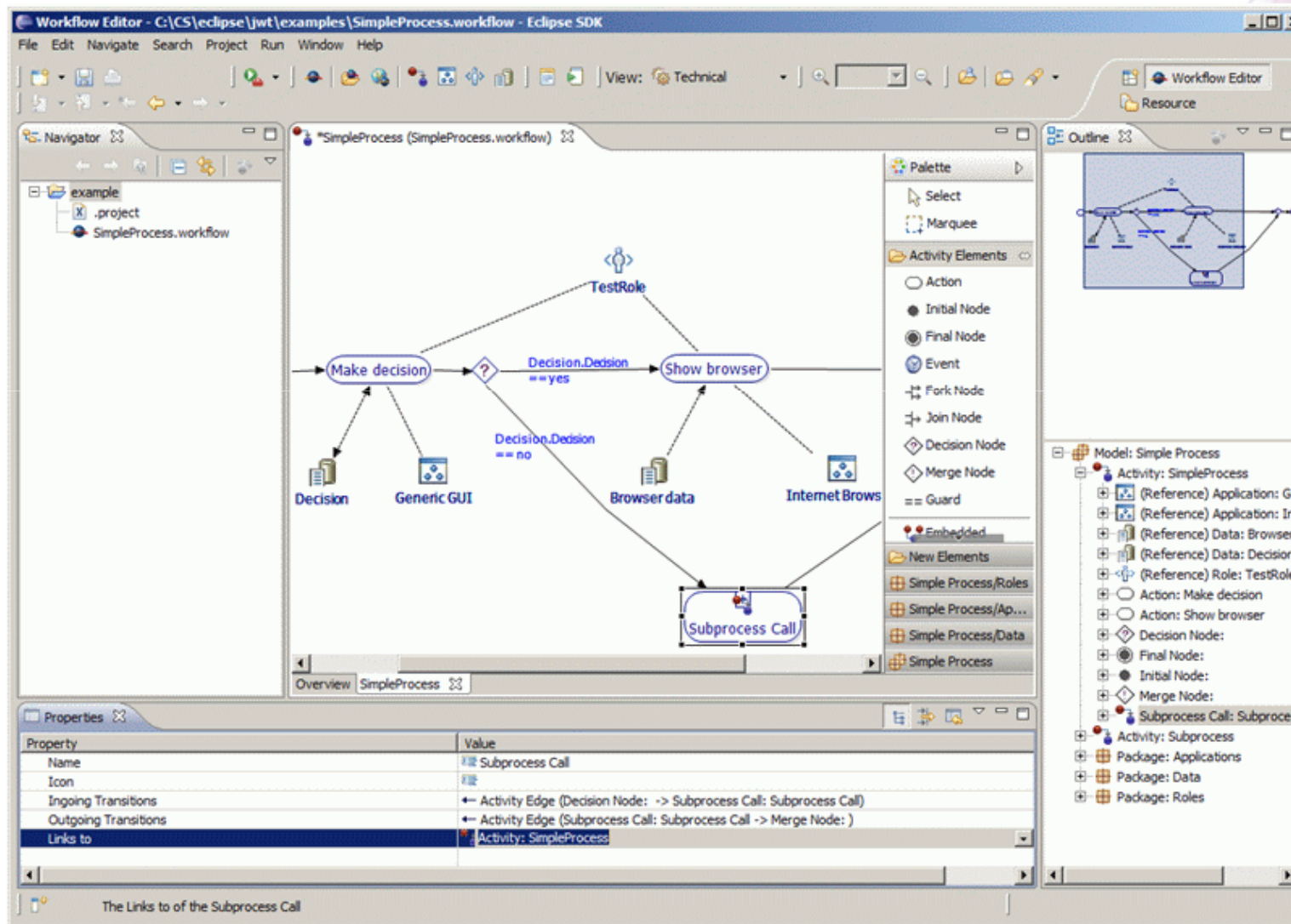
Java Workflow Tooling (JWT) aims at providing a complete Business Process Management (BPM) and workflow tooling platform

A special focus lies on a unified approach to BPM design, allowing to bridge the gap between different process representations, standards, engines and deployment environments (platforms, information systems, SOA)

Goals

- Support for workflow and business process modeling, deployment, execution and monitoring inside Eclipse
- Provide a complete, flexible, interoperable and usable BPM toolkit
- Set of generic and extensible plugins and APIs
- Extensions allowing support for specific business representations, process language formats, process engines, service platforms, etc.
- Targeting and supporting SOA in close collaboration with the former Eclipse STP projects

JWT Screenshot

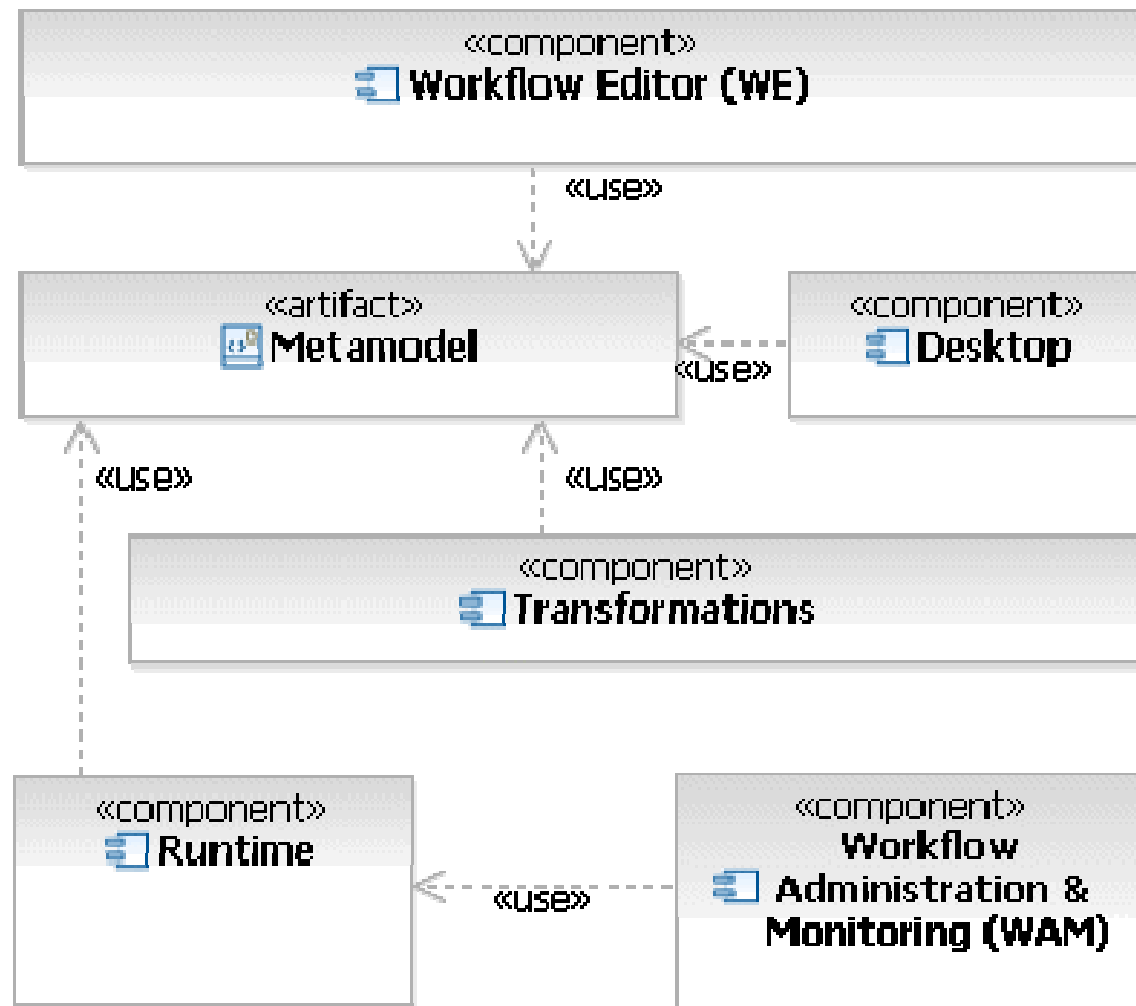


About JWT – Tool Usability



- JWT enables a user to model his/her processes and workflows and use these models not only for documentation, but also for execution.
- The model can be used to generate code in different languages (such as XPDL)
- The model can be transformed to other models (e.g. STP BPMN).
- It already provides several extension points where others can build on it and extend several parts of the editor.
- If necessary, the meta-model can be adapted to the needs of each user.

About JWT - Components of JWT



About JWT - Status



- JWT is a mature project in Eclipse SOA, already part of the yearly release train since Galileo
- The usage of JWT (Workflow Editor, Transformations, ...) is well documented in its Wiki
- Several pages on the wiki provide developer documentation. All extension possibilities by external plugins are documented there, and have examples that are available at least in the CVS.
- Ongoing development and active mailing list
- Currently in Incubation phase, graduation review and release is planned alongside Helios release train

About JWT - Maturity



- The build process is automatized for continuous integration and runs JUnit tests
- Since Galileo in the yearly release train
- Well-documented in Wiki
- Several example plugins are available that demonstrate the usage
- JWT has been integrated in several third-party solutions like e.g. Scarbo or AgilPro

Release 1.0 – Features & BugFixes



- No major new features have been added since the last release (0.7.0)
- Instead we focused on fixing all bugs still present in 0.7.0 which would complicate the everyday use of the Tooling based on the experience gained with 0.7.0 and refactored the codebase to improve the architecture

=> JWT has reached a stable and mature state

Release 1.0 – APIs



- The code conforms to the Eclipse Quality.
- The code has been implemented by committers of the project.
- The API has been officially declared (using PDE tooling) in prior to this release.
- Several examples have been tested and several junit tests exist.
- Build (includes unit testing) has been fully automated and is easily repeatable.

Release 1.0 – Non-Code Aspects



- Documentation, communication and visibility is constantly being improved
- Wiki pages dealing with changed aspects have been updated
- The documentation is up to date
- Mailing lists and forum are constantly used
- Until this date, two articles about JWT have been published in the german „Eclipse Magazin“
- Presentations about or involving JWT were given at Eclipse Summit Europe 07, EclipseCon08, Open World Forum 2008, EclipseCon09, Solutions Linux 2009, Eclipse Forum Europe 2009, Eclipse Europe Summit 2009 and EclipseCon10

Release 1.0 – Architectural issues



- Several extension points available that are already used by plugins in order to extend the JWT Workflow Editor.



Release 1.0 – Tool usability



- JWT enables a user to model his/her processes and workflows and use these models not only for documentation, but also for execution.
- The model can be used to generate code in different languages (such as XPDL)
- The model can be transformed to other models (e.g. STP BPMN).
- It already provides several extension points where others can build on it and extend several parts of the editor.
- If necessary, the meta-model can be adapted to the needs of each user.

Release 1.0 – End-of-Life



- There are no features that are end-of-life'd in this release.

Release 1.0 – Bugzilla



- There are currently 37 non-critical bugs / feature requests which will be moved to the next release

		Version	
		1.0.0	Total
Component	Desktop	<u>1</u>	<u>1</u>
	Metamodel	<u>3</u>	<u>3</u>
	Transformations	<u>5</u>	<u>5</u>
	WE	<u>28</u>	<u>28</u>
	Total	<u>37</u>	<u>37</u>

Standards



The relationship to existing standards has been achieved by the following:

- Addition of a UML Activity diagram view
 - Addition of a EPC model view
 - Transformation to the Eclipse STP BPMN modeler
 - Codegeneration of XPDL possible
-
- Existing transformation to the STP-IM (now Mangrove), which aims to bridge different standard oriented tools within the STP project, like BPMN and SCA editors or BPEL.

UI Usability



- The Workflow Editor supported originally several languages such as English, German and French. For the latest releases we collaborated with Babel so that much more languages now exist.
- Several wizards exist to ease the usage of the Workflow Editor.
- All transformations are easily selectable via Import and Export menus thanks to the transformation base.
- The Eclipse User Interface Guidelines have been followed.
- A UI walkthrough is planned for the following release

Communities



- JWT's mentor is John Graham (RedHat), former DTP Lead
- Active bugzilla usage by the committers and others (extenders).
- Many discussions on the mailing list, inside JWT as well as with partners from the STP projects (Mangrove, BPMN, etc.).
- Discussions with several companies (Bull, jBoss) have produced collaboration (ex. BPM model comparison, FactoryRegistry) and keep feeding JWT's requirements.
- JWT Integrations include AgilPro and OW2 Scarbo (Open Source). Mailing list exchanges show new products being built on JWT.
- Coordination with several OW2 projects such as Bonita, Orchestra, FraSCAti, Spagic, Scarbo.
- The University of Augsburg is an official Eclipse Foundation Member

IP Issues



- The code has been committed by individuals who are either committers of the project themselves or their foundation. Code contributions from others have undergone a CQ.
- The contribution questionnaires for all components have been completed.
- The legal information has been inserted into the source code as described in the Eclipse IP Policy.
- There are no outstanding CQs and the IP log has been committed and approved for this release.

Project plan and IP log



- The current project plan can be found at
 - <http://www.eclipse.org/projects/project-plan.php?projectid=technology.jwt>
- The IP log can be found at
 - http://www.eclipse.org/projects/ip_log.php?projectid=technology.jwt
 - Currently no open IP requests

Communication



- **Project Leads**
 - Marc Dutoo, Open Wide
 - Christian Saad, University of Augsburg (new co-lead)
 - Florian Lautenbacher, University of Augsburg (former co-lead)
- **Communication Channel**
 - JWT Mailing List: jwt-dev@eclipse.org