THALES

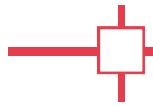
EGF Creation Review

Benoît Langlois - Thales/TCS/EPM

April 22, 2009

Communication Channel URL:
http://www.eclipse.org/newsportal/thread.php?group=eclipse.egf

Agenda



- Executive Summary
- Requirements
- Mentors
- Committers
- Best Practices
- Community Response
- Internal Implementation Focus
- Eclipse Development Process
- Future Directions

THALES

Executive Summary

- The EGF (Eclipse Generation Factories) component is a proposed open source project under the **EMFT project** to provide a **model-based generation framework**
- **Objective**: Supporting complex, large-scale and customizable generations
- In order to meet this objective, EGF:
  - Does not provide a new transformation engine or DSL editor

    ⇒ Reusing them
  - Provides an **extensible generation structure**
    - Keywords: Factory Component, Viewpoint, Production Plan (a.k.a., Orchestration). Cf. Appendix for definition.
  - Provides **mechanisms of generation customization**
    - Keywords: generation pattern. Cf. Appendix for definition.
  - Promotes the constitution of **factory portfolios** in order to capitalize on generation solutions

THALES

Requirements 1/2

- **Enough Developers**
  - The EGF project has sufficient (four) committed developers
- **Clear and Concise Description**
  - No feedback from the community of confusing or unclear terms, objectives, and boundaries of the proposal
- **Collaboration**
  - Generation results: Generation can target modeling tools (e.g., EMF) or further
  - Tools used during the generation: Generation patterns support Jet; targeted Jet-2, Acceleo
  - Tools used for the orchestration:
    - Expected: Usage of BPMN2 to express complex orchestrations. By generation, target a concrete syntax
    - Possibility to integrate other kinds of orchestration by extensibility, including its own execution mode, e.g. MWE, JWT

Requirements 2/2

- **Extensible Frameworks and Exemplary Tools**
  - Structural aspect:
    - EGF provides a default FC structure
    - A FC is an extensible structure by model extensions
      - Ability to extend the viewpoint and orchestration structure in order to add one's generation data organization, and orchestration representation with its execution mode
  - Generation execution aspect:
    - Primary execution mode, by Java task, in order to be open
    - Generation execution by FC execution
      - By FC assembly (encapsulation), ability to customize the FC execution behaviour
- **Sufficient Time for the Community**
  - The EGF project was declared on February 2, 2009

THALES

Mentors

- **Ed Merks**, Macro Modeling Inc., Eclipse Modeling Project co-lead and Eclipse Modeling Framework project lead

- **Wayne Beaton** is the Evangelist for the Eclipse Foundation, the editor-in-chief of Eclipse Corner, PMC Lead for the Technology Project, Project Lead for the Examples Project, co-Project Lead for the SOC Project, and an advisor for osbootcamp

THALES

Committers

- **Benoît Langlois**, Thales/TCS/EPM, project lead
- **Xavier Maysonnave**, Thales/TCS/EPM
- **Laurent Goubet**, Obeo
- **Thomas Guiu**, Soyatec

THALES

Committers – Benoit Langlois

Benoit Langlois is a software architect and project leader at Thales. He holds a computer science degree from EPITA (France). He has 14+ years of experience in model-driven development in small- and large-scale projects. He contributed to OMG's standards (Query / Views / Transformations, UML Profile for Modeling Quality of Fault Tolerance Characteristics and Mechanisms) and European Projects (Master, Modelware). He participated to the first steps of the ATL language.

THALES

Committers – Xavier Maysonnave

Xavier Maysonnave, currently Eclipse Expert for Thales EPM, has more than 20 years' experience developing software for enterprise architecture. He holds Bachelor of Computer Science from Paris X Nanterre University. He has worked in many different industries including aeronautic, defense and petrochemical. He used various environments from micro computers to calculators and mainframes. He applied various techniques from standalone, two tiers, three tiers, middleware and object mapping applications. He started as a Clipper developer, then a C and C++ developer, since 1997 he mainly focus on Java. His commitment to the Eclipse community started in 2002 as a founder of Omondo. He delivered EclipseDatabase, a database solution for Eclipse based on Apache Torque. Since then he strongly focused on Modeling and code generation.

THALES

Committers – Laurent Goubet

Laurent Goubet, Research Engineer at Obeo, works his daily job on software evolution and Eclipse plug-in development as a committer on EMF Compare (model comparison) and Acceleo (Implementation of the model-to-text OMG standard "MOF Model-to-text Transformation Language").
His work at Obeo involves consultancy and expertise in setting up a pragmatic and efficient model-based development processes for software providers or administrations leveraging the Eclipse Modeling technologies (EMF, EMFT, M2T...).
He graduated the University Institute of Technology of Nantes in the field of Computer Science before graduating a bachelor's degree. He then specialized himself in software engineering and model driven engineering.

THALES

Committers – Thomas Guiu

Thomas Guiu, currently eUML2 Lead Developer for Soyatec, has more than 9 years of computer programming expertise. He holds an engineering diploma from ESIEA in Paris. He has always worked for Independent Software Vendor and was involved in various technologies like Jonas, JDO and Eclipse. His commitment to the Eclipse community started in 2002 as a founder of Omondo. He was a key developer of EclipseUML. Thomas is now interested in declarative UI as developer of XWT and a proposed committer on the PMF (Presentation Modeling Framework) proposal of the EMFT sub-project and a proposed committer on the SLDT (Silverlight Development Toolkit) proposal

Best Practices

- **Communities**
  - The EGF project has a number of contributors and users to use EGF outside the core developers, Crescendo Technologies, HiPeS, Skyway Software, Soyatec
- **Diversity**
  - Contributors and interested parties work on different activities (tool provider, industry, consultancy & project subcontracting)
- **Technical Scope**
  - EGF provides an extensible generation structure with examples (e.g., generation pattern) in order how to understand and extend it
- **Maturity Plan**
  - It is expected that the EGF project will take two years (releases) to mature
- **Following Eclipse Rules**
  - The project lead and initial committers for the EGF project understand their responsibility to respect the rules of the Eclipse Foundation

Community Response

- Few but very positive feedback from the community
- The EGF project has attracted two new interested parties since its declaration
- Publication: An Introduction to Eclipse Generation Factories on Eclipse Zone of DZone (about 2,500 views)

THALES

Initial Implementation Focus

- Step 1. Initial commit (expected in July 2009)
  - Providing the default structure of a Factory Component
  - Ability to edit (properties and FC assembly) and execute a Factory Component
  - Providing the Generation Pattern solution, Jet-based
  - Application of the Generation Patterns to the EMF generation
- Step 2. Improvement of the initial commit
  - Providing the Factory Component extensibility (viewpoint, orchestration)
  - Improving the EMF generation with generation patterns
  - Supporting another engine than Jet for the Generation Pattern solution, e.g. Acceleo
- Next Step
  - Supporting complex orchestrations (e.g., BPMN-like)
- Wiki
  - Providing examples of generation with Factory Components

Eclipse Development Process

- The EGF project members and new candidates have read and understand the Eclipse Development Process and their responsibilities towards the community and especially in regards to IP issues

Future Directions

- Beyond the default EGF execution mode, the EGF project will target a workflow engine from other kinds of orchestration, such as JWT or MWE.
- The EGF project will encourage synergy with other Eclipse components, for instance to extend the default FC viewpoints and to enrich the default EGF portfolio. The driver will be the provision of valuable Eclipse generation solutions.

Appendix – EGF Terminology

- **Factory Component** (FC): Deployable unit of generation. A FC contains a set of viewpoints and a production plan.
- **Portfolio**: Cohesive and valuable set of FCs for a development team.
- **Viewpoint**: Data set describing a generation concern. A viewpoint conforms to a language.
- **Production Plan**: Executable description of a generation orchestration.
- **Generation Pattern**: Generation solution for a recurrent generation problem. A pattern has parameters, conditions, and an implementation. For the implementation, EGF proposes by default a model-to-text transformation.

THALES