March 31, 2009

Extended Editing Framework (EEF) – Creation Review

Introduction

The goal of EEF is to improve the EMF model creation phase by providing new services dedicated to editing models and creating more appealing edition elements. The way to obtain these services and elements is based on a generative approach similar to the EMF.Edit one.

Comments should be made on the EMFT newsgroup: news://news.eclipse.org/eclipse.technology.emft

Scope

The EEF project will focus on the tooling for the production of enhanced environments dedicated to the edition of EMF models. This will be obtained by three means :

- **Providing the widgets and architecture improving the user experience while editing EMF models.**
  The Extended Editing Framework runtime contains a set of advanced widgets and a generic and extensible MVC architecture that eases the edition of EMF models. The runtime also includes an extensible architecture to ease the use of these widgets.

- **Providing a way to model graphical components and views.**
  The Extended Editing Framework also comes with standard metamodels. These allow the definition of models that will parameterize the actual editing components that are to be generated. For example, any given EMF element can be related to one or more editing views. Another example is the ability to define with these models which kind of widget will be used to edit the properties of an EMF element.

- **Exemplary Tools.**
  EEF provides a set of actions and editors assisting the definition of the EEF models.

- **Extensibility.**
  EEF provides API for consumers to extends the EEF models with custom widgets and generate the corresponding code.

- **Provide a sample generation**
  The framework provides default generators for the editing components. These are organized as an Acceleo module that generates a standard architecture extending the framework's runtime. The result of the generation can then be included in standard SWT/JFace elements (composites, views, editors, …)

Out of scope

EEF is not a model repository for all and any UI models. The goal of this project is to provide modeling tools for UI generation, not the models for specific use cases.
EEF models for standard metamodel like UML or Ecore could be provided by their associated project : Papyrus, ... but will not be designed by the EEF committers.

The EEF project doesn't aim at generating user interfaces dedicated to others goals than EMF model edition. EEF specific widgets and generators are designed only to edit EMF models.

Relations to other Eclipse project

- **EMF**
  Since the project aims at providing components to edit EMF models, EEF will be build on top of EMF, especially on the EMF.Edit part of this framework.
  When EEF reaches its 1.0 release, it will get integrated in the EMF Project.

- **EMF Validation and Transaction**
  The common way to describe business rules on a metamodel is currently to define validation rules on it. In order to respect this practice, EEF will be able to interact with EMF Validation and notify the user when he tries to edit a model in a wrong way.
  The Transaction component also provides some interesting features that will be included in the EEF implementation.

- **EMF Databinding**
  The controllers generated by EEF are intended to realize a data binding between the EEF views and the associated EMF models. This could be performed by EMF Databinding in a future implementation ...

Community involvement

- The CDO project team wants to use EEF to define a vision for re-usable UI components. this also include an RCP application to demonstrate the core features of CDO.
- Itemis AG. suggests a 2009 GsoC in the scope of EEF.
- Benjamin Cabé (Anyware) wants to contribute a Forms-based editing framework which comes with facilities to deal with user-friendly error reporting.
- Alain Picard (apicard@tampabay.rr.com) is interested in creating a multi-level preference and properties setup but for a user-defined number of levels.

Code contribution

- EEF will have an initial contribution of a set of plug-ins developed by the initial team and not depending on any third party library.

Interested parties

Two new interested parties want to join the project since the original proposal :

- Benjamin Cabé (Anyware)
- Bernd Kolb (SAP)

Mentors

- Ed Merks (Macro Modeling, Canada)
  PMC lead of Eclipse Modeling, Project lead of EMF
- Cédric Brun (Obeo, France)
  Member of the Planning and Architecture council, Project lead of EMF Compare

Initial Committers

- Goulwen Le Fur (Obeo) - Project Lead
  Goulwen works as MDA consultant and software engineer in Obeo. The EMF and GMF frameworks are included in most of the projects he works on in Obeo. He defined the prototype of the EEF project based on the requirements of projects like Papyrus (properties views generation) or industrial MDA project for the « Pôle Emploi » (Enhanced user interfaces production for the Pôle Emploi Domain Specific Models).

- Stéphane Bouchet (Obeo) – Committer
  Stéphane works as MDA consultant and software engineer in Obeo. He works on many industrial projects based on several Eclipse components like EMF, GEF, BIRT ... He used his experience in the user interface design and in the code generation (Acceleo) to improve the EEF project, especially in the generative part of the project.

- Nathalie Lépine (Obeo) – Committer
  Nathalie works as MDA consultant and software engineer in Obeo. She works in many projects based on both the EMF and GMF frameworks (industrial MDA tools for code generation, weaving model exploitation, domain specific modelers design ...). She improved the runtime of the EEF project for customers purpose (advanced widgets design, ...).

- Patrick Tessier (CEA List) – Committer
  Patrick Tessier works for the CEA in order to produce high quality Eclipse based tools. He is a committer on the Papyrus project on Eclipse and he contributed to the definition of the editing components design.

Roadmap

- The Creation Review will take place on April 8, 2009
- A first release of the EEF project (v0.7.1) will be made in May 2009 since the Papyrus project already uses it. This first release will contain all the elements needed to produce properties views as used in the papyrus project. A first version of the editors, generators and widgets will be included in this version.
- A second release (v0.8.0) will take place in December 2009 providing an alternative generation extending the EMF.Edit one
- We plan to join the release train for the Eclipse 3.6 simultaneous release with a v0.8.1 providing more widgets and a consolidated architecture with EMF Databinding.

Copyright Statement