Creation Review for the Proposed PMF Project

Planned review date of April 22

Yves YANG, Jim van DAM
April 1st 2009

1. Background

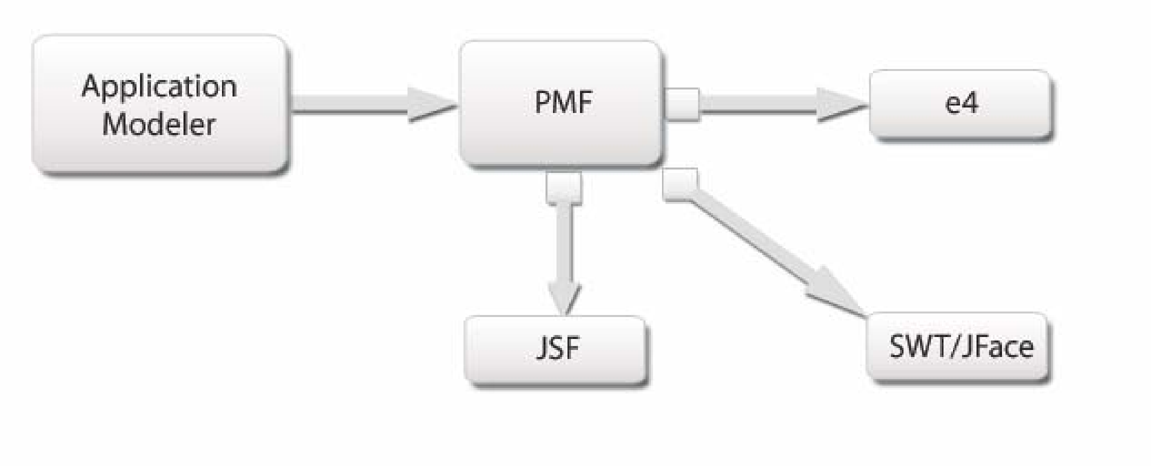
The Graphical User Interface (GUI) is the visible part of the software. It really consists of the gate through which business features are made available and it reflects the application quality. Designing the visual composition and temporal behavior of GUI is an important part of software application programming. Its goal is to enhance the efficiency and ease of use for the underlying logical design. Typically, the user interacts with information by manipulating visual widgets that allow for interactions appropriate to the kind of data they hold. The widgets of a well-designed interface are selected to support the actions necessary to achieve the goals of the user.

Model-Driven Engineering is a modern, standard and industry software engineering. It refers to a range of development approaches that are based on the use of software modeling as a primary form of expression. Sometimes models are constructed to a certain level of detail, and then code is written by hand in a separate step. Sometimes complete models are built including executable actions. Model-driven architecture supports model-driven engineering of software systems. MDA provides a set of guidelines for structuring specifications expressed as models. The MDA approach defines system functionality using a platform-independent model (PIM) using an appropriate domain-specific language (DSL).

Obviously, MDE in GUI needs an appropriate framework to simplify the presentation generation. It is in fact a critical component in scalable enterprise data presentation applications such as ERP and BPMS in Manufacturing, Supply Chain, Financials, Customer Relationship Management (CRM), Human Resources, Warehouse Management and Decision Support System.

From users' perspective, the UI presentation modeling should be composed of following steps:

- Business modeling
- Importation of the business model and initialization of PMF structure
- High level UI presentation modeling in PMF editor
- Model transformation from PMF model to displaying technology execution code



PMF focuses in fact on high level presentation modeling concepts on GUI by ignoring the displaying technology artifacts such as appearance, layouts and data binding support. It allows each technology to make its specific connection with this framework.

Comments should be made on the Modeling newsgroup:
news://news.eclipse.org/eclipse.modeling

2. Scope

The purpose of this framework is to provide the basic functional concepts of user interaction in a PIM level UI modeling language. The language can be extended in two ways:

- Higher level patterns can be defined using PMF itself and used as language extensions
- Transformations/Generators from the PIM level model to specific UI technology platforms

Each displaying technology such as XUL, e4, JSF, SWT, Swing, GWT, Ajax, Silverlight or others, can extend this framework to implement a specific generator for producing executable codes in Java or in declarative language.

Precisely, PMF provides following components:

- An EMF-based foundation that covers the basic functional concepts of user interaction.
- Examples of higher level patterns like Master/Detail, Finder and Selector defined in and using the basic PMF concepts
- Template engines' integration for code generation
- Two reference implementations are proposed (integrations with other UI solution will be explored as the project evolves):
  - e4's declarative UI and modeled workbench
  - SWT/JFace, being the current state of the art UI platform for Eclipse development

This framework is not a specific UI framework for a displaying technology, neither a specific UI Markup language. It is a high level modeling framework to design the data presentation scenario/pattern and user interactions.

3. Relationship with Other Eclipse Projects/Components

It is not our intention to duplicate efforts underway in other Eclipse projects. However, given the broad scope, there is potential overlap with other Eclipse community projects. Where possible and practical, we will seek reuse and collaboration.

PMF will use mostly Eclipse tools to archive the UI Presentation modeling and code generation. The most important tool is the modeling transformation engine. Several engines are used in different Eclipse projects:

- JET, a derivative of JavaServer Pages (JSP) initially developed as part of the Eclipse Modeling Framework (EMF), and currently an incubator project JET2 in the Eclipse Modeling Project.
- Xpand is a statically-typed template language with a lot of advanced features

Obvious integrations between PMF and other Eclipse components (EcoreTools, UMLTools, M2T, MTL, EMF, EMFT, VE, etc.) will be explored as the component evolves.

e4 has planed to develop a Declarative UI framework and modeled workbench based EMF. They are the perfect concrete UI solutions for testing and demonstrating the concepts of PMF.

4. Mentors

- Ed Merks (Macro Modeling)
- Dave Carver

5. Initial Committers

The initial committers for this project would be:

- **Yves Yang**, Soyatec (project co-lead) –Yves YANG, main founder of Soyatec and Committer of Eclipse VE and e4, has over 17 years of experience working with OO software development. He was the chief architect of EclipseUML (First EMF/UML native Eclipse Modeler) and co-founder/CTO of Omondo from 2002 to early 2006. In Soyatec, regarding UI programming, he has leaded and developed a first solution of XAML for Java: eFace, and contributed a Declarative UI in e4: XWT.
- **Jim van Dam**, HiPeS (project co-lead) - Jim van Dam started as a knowledge engineer in the 80's. Model based software development has been his main interest since 1985. He started with case-tools, moved on to tailor made MDSD environments and currently uses the Eclipse MDSD tooling and the DSL toolkit

from MS. He co-developed several Software Factories featuring complete application generation. His current focus is PIM-level UI Modeling and transforming these models to several target platforms.

- **Thomas Guiu**, Soyatec – Thomas has more than 6 years' experience in Eclipse development. He had developed EclipseUML from 2002 to early 2006. And then he is the principal developer of eUML2, eFace, eclipse4SL and XWT in Soyatec. He is specialized in most of the eclipse frameworks: GEF, EMF, GMF and UML2.
- **Olivier Moïses,** Generic Concept - Olivier Moises works as software consultant since 1993 and started programming with Eclipse 6 years ago (Eclipse 2.0).

  Since that time, he wrote many applications (standalone, RCP or IDE plugins) using Eclipse, organized several Eclipse RCP trainings and provides consulting services for companies which use or plan to use Eclipse. He is specialized in GEF, EMF, WTP, GMF. He is a member of OSGi&trade; Users Group France.

  Since his first release of a RCP application, he realized it should be possible to build GUI like web developers do : by using a dom like live model. He started wazaabi 1.0 an XML based declarative UI framework in 2003. It has been published in 2006 (http://www.wazaabi.org , LGPL) and is still in use in several companies. Wazaabi 2.0 (under EPL) is a whole new declarative release, based on an EMF live models and providing also stylesheet and databinding declarative models and mechanisms.

6. Code Contributions

Soyatec will offer an initial code contribution. An implementation of the PMF metamodel consisting of Java packages within the org.eclipse.pmf.* namespace will be generated and customized using EMF.

7. Interested Parties

Thus far, interest in this component has been expressed by:

- Embarcadero Technologies
- Soyatec
- HiPeS
- Lyria - group W4
- Ordina
- Springsite
- Oslo Software
- Obeo
- itemis AG
- Brane Corporation
- Tom Schindl
- Florian Lautenbacher
- Cameron Bateman
- Generic Concept

- ProxiAD
- SKHIRI Sabri

8. Developer Community

The team of initial committers will explore statements of interest from additional developers experienced with UI development and modeling or willing to gain such experience.

9. User Community

It is expected that the user community for this component will consist primarily of developers, given that it is essentially a foundation for building UI modeling tools.

10. Tentative Plan

The development plan of this project would be aligned with e4 plan. The first community technical preview release is scheduled at summer 2009 and the final release is targeted in June 2010.

11. Copyright Statement