Model Driven Development with EMF and EclipseLink
(experiences in MDD and generating user interfaces)

Suresh Krishna, Oracle Inc.
EclipseCon, 03.20.2008.
Background:

- Model acts as the “heart” of the business application

- Model changes often have considerable effect on the UI and persistence layer

- Code generation is easy but the customization takes a lot of effort
Requirements:

- Model changes should have minimal effect on the application.

- Code generation for user interface and persistence layers reduces the development effort.

- User interface must represent the domain to the closest extent possible.

- Use industry standards and well-tested ones with enterprise software quality.
Candidates:

- Naked Objects
- JMatter
- EMF and EclipseLink
Naked Objects:

- All you need to develop are your domain objects.
- The platform auto-creates an o o user interface (Swing, HTML)
- Auto-generate the persistence layer directly from the domain model definition via Hibernate
- At run-time the framework inspects the domain objects using reflection, and then renders the objects and methods visible to the user.
JMatter:

- A rich GUI built on top of Swing with support for CRUD operations.
- The GUI is constructed dynamically at runtime from the model.
- Persistence (by leveraging Hibernate)
- Authentication - the ability to login and to manage users.
- Support for the construction of wizards, calendaring and more.
Eclipse Modeling Framework (EMF):

The EMF project is a modeling framework and code generation facility for building tools and other applications based on a structured data model.

- EMF-generated code has a built-in change notification mechanism
- EMF provides a reflective API to access instances of your models and allows you to dynamically create models.
- EMF supports validation of model constraints.
- EMF provides powerful code generation tools that support regeneration of models and merging with user written code
- EMF goes RCP
- Generates ready to use User Interface on domain objects
**EclipseLink:**

This is Eclipse persistence services project which delivers an open source runtime framework by supporting several standard frameworks.

- EMF model persistence in Teneo
- Dali JPA Tooling
Teneo:

- Teneo is the eclipse sub-project from EMFT, which aims at providing the database persistency solution for EMF using Hibernate or JPOX/JDO 2.0.

- allows you to start with your model (UML or XML Schema) and automatically generate the java source code and object-relational mappings.

- takes over much of the manual work of creating relational mapping.

- The integration with EMF allows you to generate Eclipse RCP editors which persist automatically to a relational database.
Dali JPA Tools:

- It’s a WTP sub-project

- Provides support for definition, editing, and deployment of OR mapping for JPA entities

Tooling Features:

- Rich UI for entities
- JPA Validation
- Entity generation from Tables
- JPA annotation code completion
- JPA annotation and XML mapping support
Resources:

- JMatter: [http://www.jmatter.org](http://www.jmatter.org)
- Naked Objects: [http://www.nakedobjects.org](http://www.nakedobjects.org)
- EMF Home: [http://www.eclipse.org/emf](http://www.eclipse.org/emf)
- EclipseLink: [http://www.eclipse.org/eclipselink](http://www.eclipse.org/eclipselink)
- EclipseLink Userguide: [Wiki](http://www.eclipse.org/eclipselink)
- Elver/Teneo: [http://www.elver.org](http://www.elver.org)
- Dali Project: [http://www.eclipse.org/webtools/dali](http://www.eclipse.org/webtools/dali)
Thanks for your time !!!

Enjoy your MDD with EMF and EclipseLink.

Questions: suresh.madhuvarsu@oracle.com
Myself: http://www.sureshkrishna.com
Eclipse Blog: http://sureshkrishna.wordpress.com