Compuware OptimalJ standardizes on Object Management Group's Model Driven Architecture

Developing enterprise applications using Java technologies is not for the faint-hearted. Writing applications for the Java 2 Enterprise Edition (J2EE) platform has proven to be complex, difficult and tedious, slowing down advanced Java developers and creating a barrier to entry for many mainstream developers. There is a critical shortage of advanced Java developers, especially those with experience in Enterprise Java Beans (EJBs). This shortage slows time-to-market for business applications and challenges application reliability and performance. To enable developers of all skill levels to build reliable, high-performance components, the ideal solution would be to simplify Java development with a framework for delivering J2EE-compliant business applications. OptimalJ from Compuware is such a solution. OptimalJ is an advanced development environment that makes complete use of Object Management Group's (OMG's) Model Driven Architecture (MDA) to enable the rapid design, development, modification and deployment of J2EE business applications. OptimalJ generates complete, working applications directly from a visual model, using active synchronization to keep both model and code up to date during rapid application changes. It includes a built-in web server and servlet engine, J2EE application server and DBMS for unit-testing purposes, allowing the developer to fully test the application before deployment. For production purposes, OptimalJ generates archive packages and server deployment descriptors automatically, allowing application administrators to rapidly deploy J2EE applications in the target environment.

OptimalJ is one of the first advanced development environments to implement OMG's MDA in its entirety. In line with MDA there is a clear distinction between the models and the implementation of application components in OptimalJ. The models in OptimalJ have various abstraction levels, which include:

- The OptimalJ Domain Model, which maps to the MDA Platform Independent Model (PIM). The Domain Model defines the business domains without any specific application detail. It is defined by modeling the business application's functionality and behavior, domain-specific information based on the Meta Object Facility (MOF) and Common Warehouse Model (CWM) in UML.
- The Domain Model is transformed automatically into an OptimalJ Application Model, which maps to the MDA Platform Specific Model (PSM). The Application Model defines the application, based on a chosen technology, which in the case of OptimalJ is J2EE. It describes what is to be generated in order to implement the application, by providing a logical overview of the components contained within each tier that makes up the application.
- Finally, the Application Model is transformed automatically into an OptimalJ Code Model, which maps to the MDA Implementation Model. The Code Model defines the generated application code. In the case of OptimalJ, once the Web tier, EJB and DBMS tiers have been defined, the code implementations of defined components can be generated i.e. the Java classes, SQL scripts and other technical items that are needed for the components in the models.

The benefits of a solution based on MDA are enormous. MDA supports applications throughout the application life cycle from application development and design through deployment and ongoing maintenance. This offers organizations crucial return on investment, as existing applications and domain models can be reused and adopted to ongoing business and technology changes. It allows a software environment to evolve and change while the business model (PIM) remains stable. This stable base is necessary to cope with the constantly changing environment of today's market. For additional information, features and benefits of Compuware's OptimalJ advanced development environment, visit http://javacentral.compuware.com/