statement of direction

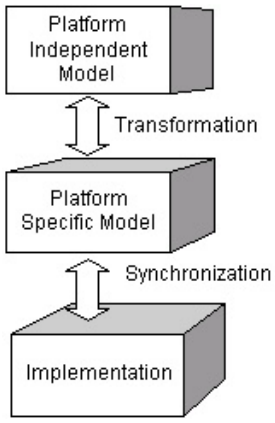
UML Modeling and MDA in Oracle JDeveloper 10*g*

August 2004

ORACLE

OVERVIEW

Oracle JDeveloper 10*g* supports the core set of UML modelers required to capture the analysis and requirements of an application in a technology neutral business model. In addition, JDeveloper supports a comprehensive set of technology focused profile modelers to express the detailed application design. These visual editors are two-way synchronized with the source and metadata files of the implementation. A Model Driven Architecture (MDA) style of working is supported by capabilities for linking and transforming between these two kinds of models, referred to as Platform Independent Models (PIMs) and Platform Specific Models (PSMs) in MDA context.



UML

The Unified Modeling Language (UML) has become the de facto standard for visual modeling. Oracle JDeveloper's strategy is to provide fully integrated modeling support for the UML modelers that developers use to capture the requirements and design of J2EE application architectures. In JDeveloper 10*g* release 9.0.5.2, support for the following diagrams is provided compliant with the UML 1.4 specification:

UML Class modeling

UML Class modeling enables the representation of the business objects of an application. These models capture the requirements of the application as the user sees them. As such, UML Class models provide an abstraction of the underlying implementation technology, which could be Java, EJB or ADF Business Components. UML Class modeling can also be used to represent package structures and application architectures.

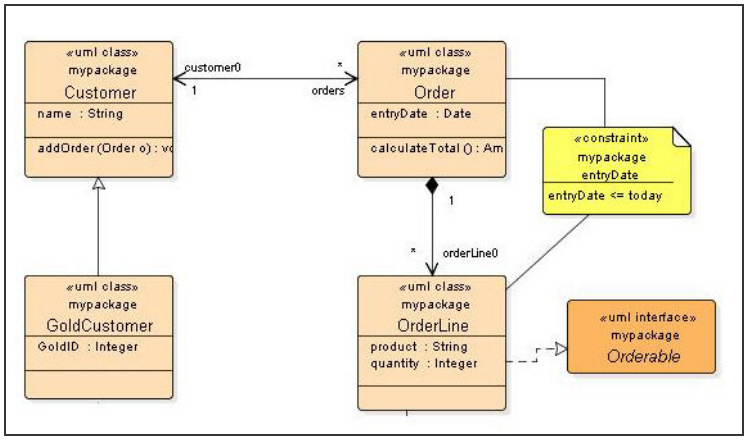


Figure 1 UML Class modeling in JDeveloper 10g for describing the business objects of an application.

UML Use Case modeling

UML Use Case modeling allows the definition of the essential services that the system exposes to the user. The graphical models are underpinned by textual details regarding the sequence of steps that the user typically goes through to achieve a successful interaction.

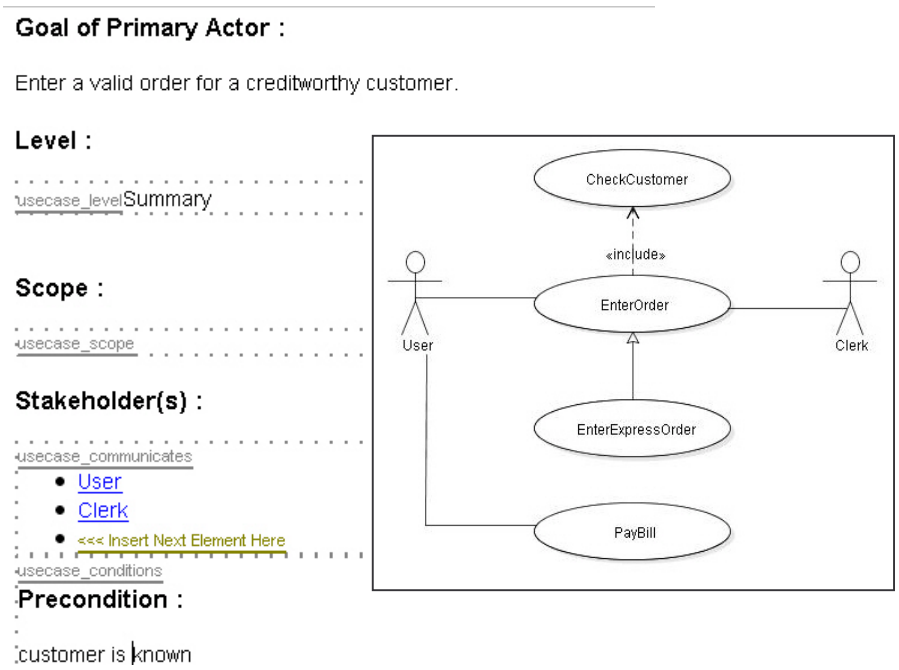


Figure 2 UML Use Case visual modeling and textual editing in JDeveloper 10g for capturing the services that an application offers to the users.

UML Activity modeling

UML Activity modeling enables the modeler to capture the business process that an application exposes. These processes are independent of their implementation, which could be Struts pageflow, BPEL integration process, or manual implementation in Java. Business processes often define the process behind a use case, or the sequencing of use cases.

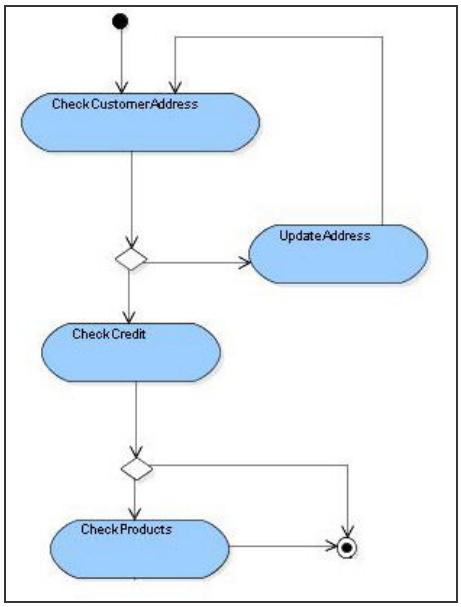


Figure 3 UML Activity modeling in JDeveloper 10g for capturing the business process of an application.

UML Sequence Modeling

Going forward, this set will be extended in JDeveloper 10g release 10.1.3 to include UML Sequence modeling. Sequence modeling describes the interactions between objects and actors in an application in terms of the flow of messages between them. These models may be constructed both at the analysis level to provide an overview of anticipated flows, or at the detailed implementation level to represent actual method calls. The initial release of this modeler will support integration with the JDeveloper debugger to automatically visualize traces; subsequent releases will add detailed code generation and reverse engineering.

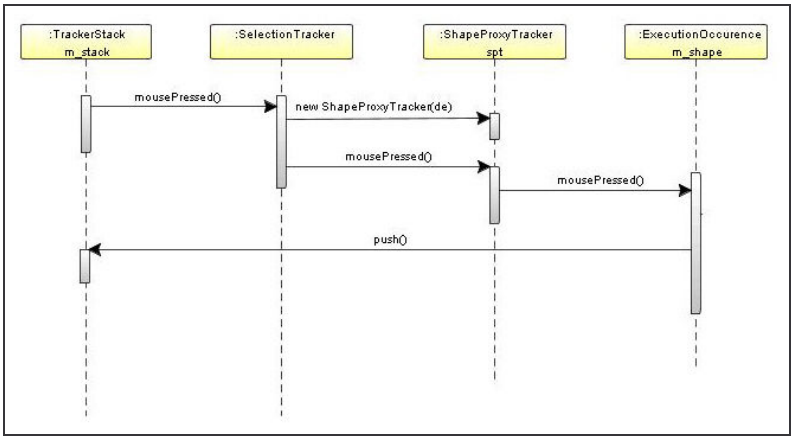


Figure 4 UML Sequence modeling will be added in JDeveloper 10g 10.1.3 to capture sequences of messages between objects and actors.

In JDeveloper 9.0.5.2, many usability and scalability enhancements have been implemented for the modelers, including enhanced autolayout of diagrams, a thumbnail overview, the publication of diagrams in Javadoc format, and the ability to insert links to other diagrams, external files, and generic URLs.

With the addition of Sequence modeling, JDeveloper 10.1.3 will support the "top 4" UML diagrams that the majority of developers use. Together, these modelers cover the complete modeling needs in approximately 80% - 90% of enterprise development projects.

Some organizations require a wider set of modelers than are supported natively in JDeveloper. For example, organizations that deploy a complete structured "unified process" approach (such as Rational Unified Process). To address these requirements, Oracle is pursuing a strategy of partnering to provide a complete UML modeling solution within JDeveloper, including:

- Visual Paradigm, SDE
- CanyonBlue, Kanesa
- Softeam, Objecteering

More information about these partner products can be found on the JDeveloper extensions page on OTN (http://otn.oracle.com/products/jdev/htdocs/partners/index.html).

XMI support

Some organizations may want to retain their existing UML modeling tool and use it in conjunction with JDeveloper for downstream development. To enable this, JDeveloper 10g 10.1.3 will provide XMI export of class models (compliant with XMI 1.1) to complement the current XMI import. Class diagrams capture the core information that is needed for code generation. By importing these models into JDeveloper, code generation and design updates can be made using JDeveloper, after which the updated model can be exported for inclusion in the external modeling tool, where any required updates to the wider UML model can be made.

UML 2.0 support

During 2004, the OMG is expected to finalize the new UML 2.0 specification, currently available in draft form. Oracle will upgrade its modeling support in JDeveloper to become compliant with the new 2.0 standard. In addition, the new "composite structure" diagrams are an area that aids architectural modeling as well as component definition and wiring. Some of these component aspects are already supported in JDeveloper 10g release 9.0.5.2 for modeling Application Modules in the Business Component modeler. Now that the UML 2.0 standard incorporates these aspects, this will be a focus area going forward.

TECHNOLOGY SPECIFIC MODELERS

Technology specific modelers (sometimes referred to as "platform specific" modelers or "UML profile" modelers) are visual editors that let the developer modify source code and meta data in a declarative, visual manner. Source code and visual model are permanently synchronized: changes in one view are immediately propagated to the other. The use of profile modelers makes J2EE technology easier to access for non-experts, makes the development process more productive, and provides more effective communication between team members. The majority of UML modeling tools support profiles in a generic manner, as annotations on basic UML models. However, Oracle JDeveloper goes one step further, providing 'native' support for profiles with specialized notation, dialogs and wizards. In JDeveloper 10*g*, the following profile modelers are supported:

- Java modeler
- EJB modeler
- Business Component modeler
- Database modeler
- Web Services modeler
- XML modeler
- Struts modeler

For Oracle JDeveloper 10g release 10.1.3, this set will be extended to include:

- BPEL process integration modeler
- Workflow modeler

Going forward, Oracle JDeveloper will provide a Modeler SDK (including samples and API) to enable third parties to develop other profile modelers and plug them into JDeveloper.

MDA

Oracle's position on the Model Driven Architecture approach to development is that it can only be effective if it enables code-oriented developers to work seamlessly with those that prefer visual representations. For instance, JDeveloper's Struts page flow modeler can be used to visually design the Struts configuration file. In addition, the visual editor will also pick up any changes from developers that prefer to hand-edit the configuration file directly. JDeveloper's comprehensive set of UML profile modelers allow visual editing to be used as one *optional* way of modifying code and meta data, in addition to wizards, property inspectors and text editing.

Furthermore, development projects that require an explicit model of the requirements of the application can optionally transform the resulting UML analysis artifacts into a technology (or platform) specific model. This is a fundamental tenet of the Model Driven Approach to development. In Oracle JDeveloper, these transformations can be re-run incrementally, an important requirement for synchronizing changes to the requirements with the implementation.

JDeveloper 10g release 9.0.5.2, supports the following transformations:

- UML class model to Java class model
- UML class model to Business Component model

Going forward, Oracle intends to extend the flexibility of mappings and incrementally add the most appropriate transformations to the Oracle Application Development Framework, specifically

- UML class model to Database model
- UML class model to EJB model

Some organizations, such as system integrators, require additional mappings and customizable MDA where the transformation rules can be adapted and extended by the developer or architect. To enable this, Oracle JDeveloper 10.1.3 will have partner solutions based on XMI import/export, including:

- Codagen, Codagen Architect
- AndroMDA (open source)

UML Modeling and MDA in Oracle JDeveloper 10*g*
August 2004
Author: Guus Ramackers

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com