

Power Modeling for Agile Developers

GMF's Fruit: Apollo for Eclipse

Apollo for Eclipse, a new visual modeling tool from the Gentleware stable, promises to ease the life of an agile developer by combining the advantages of visual modeling using UML (Unified Modeling Language) with programming in Java. Based on open source technology, developed by the Eclipse Graphical Modeling Framework (GMF) project, Apollo for Eclipse focuses on round tripping between Java and UML class diagrams, which is the core capability required by agile developers.



Introduction

Gentleware's origins are strongly rooted in open source and open community. Founded in 2000, Gentleware's interaction with the open community dates back to its work on the open source modeling tool ArgoUML and its commitment to the work of the Object Management Group (OMG) in the definition of UML 2 and the Diagram Interchange. So it came as no surprise when this Hamburg, Germany, based company recently announced its truck with the Eclipse foundation. Its first offering, Apollo for Eclipse, is a new visual modeling tool that is tightly integrated with the Eclipse development environment (IDE), and primarily aimed at agile developers who want to flexibly combine the advantages of visual modeling using UML (Unified Modeling Language) with programming in Java.

Target Groups

- Code-driven agile developers interested in mechanisms to quickly access the code in a model representation and gain additional insights into its structure. Look out for support this style of work with additional features in the next releases.
- Model-driven developers that are interested in increasing productivity by leveraging model-to-model or model to code transformations. These needs require a good understanding of the environment where it is employed.

Apollo for Eclipse addresses the modeling needs of programmers & developers. For example, if you are assigned with the task of understand existing code, Apollo for Eclipse reveals the structure and makes it accessible in class diagrams. Apollo for Eclipse can also quickly provide you with the diagrams you need, should you need to document the code that you have just developed. It also provides a way to navigate and extend the code while you are working on it. The roundtrip capabilities of Apollo have a lot of usage scenarios

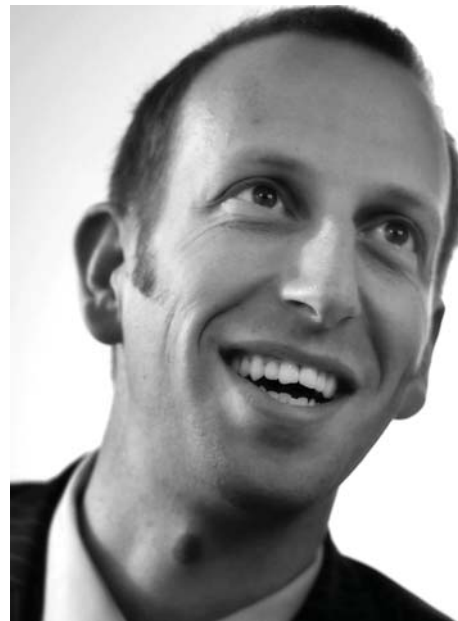
Getting Started

The easiest way to get started with Apollo for Eclipse is to use the Rich Client Platform (RCP). Download the RCP zip file from the Gentleware web site, unpack the file to the desired directory, and run Apollo for UML from this directory. Apollo for Eclipse may also be added as an extension to the Eclipse IDE. It requires some features to be installed to Eclipse in addition to the Apollo for Eclipse extension. The simplest way to do this is using Eclipse Callisto, although you may also choose to install each part manually.

If you decide to use Eclipse Callisto, which we strongly recommend, your installation is limited to three simple steps:

Feature	Version Number
eclipse-SDK	3.2.1
eclipse-test-framework	3.2.1
eclipse-RCP-delta-packt	3.2.1
emf-sdo-xsd-SDK	2.2.1
emft-ocl	1.0.1
emft-query	1.0.1
emft-transaction	1.0.1
eclipse-test-framework	3.2.1
emft-validation	1.0.1
GMF-sdk	1.0.1 with patch
apache-derby	10.1.2.1
GMF-sdk	1.0.1 with patch
uml-SDK	2.0.1

Table 1: Eclipse Features Required for Apollo for Eclipse



“Apollo for Eclipse Combines Modeling with Coding”

Dr. Marko Boger, CEO, Gentleware

Step 1: Make an update through Eclipse Callisto: after the installation is complete, shutdown Eclipse.

Step 2: Install the UML2 Eclipse project: you can download the UML2 project from the Eclipse site (<http://www.eclipse.org/uml2/>)

Step 3: Install Apollo for Eclipse: you can download Apollo for Eclipse from the Gentleware web site (<http://www.gentleware.com/downloadcenter.0.html>).

Step 4: Start Eclipse

However, if you decide to install the required features manually, see **Table 1** below for the list of necessary features. The rest of the steps (**Step 2** through **Step 4**) remain the same as above.

Feature Breakdown

Apollo for Eclipse is directly built on the frameworks of Eclipse and integrates very tightly with Eclipse. Some of the features in version 1.x include:

- Forward, reverse, and roundtrip engineering
- Support for Java 5
- Support for UML 2.1
- Class diagram handling
- Available as a standalone RCP or as an extension to Eclipse

A diagram view of a given class or package is just a click away, and the code corresponding to a diagram element is also right at your fingertips. “The most impressive feature is that the roundtrip engineering is so seamless and leaves no traces in the code. And it is lightning fast and scales very well. We have used it on very large projects with thousands of classes and have found no performance or scalability problems”, says Dr. Marko Boger, CEO of Gentleware.

An eye-catcher is the automatic layout feature that produces very readable diagrams. When you initiate the layout, the animated morphing helps to keep track of where your classes are. The filtering functionality also helps you focus on just what you want to show in your diagram.

Apollo for Eclipse has fully accessible APIs as it is built on open source frameworks. It makes use of the underlying frameworks of the Eclipse Modeling Framework (EMF). At the core of this is EMF, which holds the model information at runtime. The ability to fully access that via the API allows a tight integration with other tools, like code generators or model-to-model transformations, metric tools, validators, consistency checkers and so forth.

The initial release focuses on the class diagram, with a lot of emphasis on making this diagram type as useful as possible, for example with the fully synchronized roundtrip engineering capabilities. “Other diagrams will follow shortly. But before we go into the breadth of the entire UML, we want to make sure to provide the highest possible gain from the most important diagram type of all, the class diagram”, says Boger.

Catching Up with Trends In the Model-driven Development Space

As model-driven development technologies have become mainstream, several trends have become prominent. These trends include agile modeling, domain-specific modeling, and open source development. “Gentleware’s new Apollo for Eclipse product line embraces all three of these trends. First, it supports agile modeling since it uses a pragmatic subset of UML 2.1 that supports effective Java roundtripping. Second, it facilitates domain-specific modeling because it is designed to be customized and extended for specific domains using the

UML 2.1 profile mechanism and the Eclipse framework. Lastly, it supports open source development since the new product is based on Eclipse GMF, the best open source modeling technology currently available”, said Cris Kobryn, a Gentleware board member, who also chaired the UML 1.1, UML 2.0 and SysML standardization teams. Boger also highlighted the increasing importance of Eclipse and its frameworks as de-facto standards for integrating the various tools involved in software development across the entire life cycle. “Apollo for Eclipse fills a gap here at the very beginning of the tool chain for MDD. Then there is the trend towards a more agile modeling style. For that, diagrams need to directly reflect the current state of the code and vice versa. The roundtrip capabilities of Apollo for Eclipse fulfill that. And then there is the trend towards domain specific modeling. Apollo for Eclipse is an instance of Gentleware’s capability to build modeling tools. UML really has become an example of a specific notation and language definition. We can quickly build custom tools for a variety of modeling notations or languages. Apollo for Eclipse proves our ability to do so in a most efficient manner”, he says.

End Analysis

Apollo for Eclipse combines modeling with coding. For anyone who knows the UI of Eclipse a bit, it is a snap to use. It helps to understand the complexities of code and makes code more digestible. “It nicely confirms our analysis that a good roundtrip solution for Eclipse was missing. With a very low price point, starting at 5 Euros, we are addressing a very large potential customer base. While high-end modeling tools with a coding and modeling integration used to be very expensive and excluded large parts of the developer community, for the first time we now can provide this quality to the full range of users, from students to large enterprises.”, says Boger.

Resources & References

[1] <http://www.gentleware.com/>

[2] <http://www.gentleware.com/apollo.html>