



Code Generate!

Improve your chances of success and reap the return on investment with code generation & reuse of company assets.

Map your application to MDA

Systems are future-proofed by following the OMG's Model Driven Architecture (MDA) approach. Map your models to a chosen CodeGenie implementation environment.

Analyse your application

Don't just draw throw-away design diagrams in UML. Analyse the problem and produce implementation-free diagrams that define the system.

What is a Platform Independent Model (PIM)?

Business analysts document the result of their analysis in a subset of UML called eXecutable UML (xUML). As its name suggests the xUML models is an executable model, but it is defined with regard to the business needs, with no operation environment details. Whilst some tools that can test xUML models directly exist, the most common mode of implementation is to pass the Platform Independent Model through a MDA compliant tool such as CodeGenie to generate code in a 3rd generation language such as Java. This process takes seconds.

What is CodeGenie?

Domain Solutions' CodeGenie turns models into code. It supports the Executable UML process within the OMG's MDA approach for automatic code generation from UML analysis models. CodeGenie is open and can accommodate the client's software processes, target software architecture and software development tools.

What is an Architecture Model?

An architecture model describes the target deployment environment. It covers issues such as distribution, communications, persistence, deployment etc. CodeGenie takes the architectural model, which is written in executable UML and generates a new version of itself for that particular architecture. It's easy to make architecture changes for new and improved design patterns. Either use an off-the-shelf architecture e.g. Java J2EE or define your own.

Why bother?

The potential reduction in code size when using a MDA tool can be enormous, as the graphics to the right illustrates. In the quoted study only 250 lines were in conventional code, the remaining 750 were textual representations of the UML diagrams used to describe the system. The potential cost savings resulting from this reduction in complexity are huge. In a direct comparative study carried out by www.middleware.com where the same system was developed by two equally experienced teams, manpower savings of the order of 35% were



realised in favour of the MDA approach, even without making any allowance for the learning curve involved. Still greater savings are expected during the maintenance of systems. Plus you safeguard IPR whilst understanding your projects and software architecture.

Sample Architecture – J2EE

J2EE is short for Java 2 Enterprise Edition. It is a n-tier architecture which is normally used for implementing large web based systems (see illustration below). Introduced around 5 years ago, best practice is still being developed and documented as design patterns. Although J2EE is defined by Sun, there is such a variety of implementation options that the term really describes potentially hundreds of different architectures.

What does the architecture consist of?

The primary artefact is a J2EE implementation architecture model in UML. This model covers all of the deployment artefacts required to fully implement an application system. This includes SQL to create an initial database, Entity Beans, Session Beans, Servlets, JSP pages, DTO's, any J2EE deployment files and ANT build files. In addition it delivers a populator which translates an XMI (XML Metadata Interchange) representation of the application model into a Platform Specific Model (PSM) and also consists of code generation archetype scripts which generate Java source code from the PSM.

Is it just for Java?

No, far from it! The method has its origins in the Real Time/Embedded market with hard constraints for size, speed and reliability. This approach is efficient and repeatable.

Case Study: AMS (BAE Systems) Radar Controller

- Industrial Strength architecture built on repeating design patterns.
- Built on the OMG CORBA platform offering object distribution
- Generates around 80% of the system in IDL/C++
- Several hundred thousand lines of code
- Metric show return on investment already gained on the first iteration

Case Study: EuroControl ATC Course Management System

- Air Traffic Control training system with external scenario/results management
- Built on the Java J2EE platform
- Generates HTML user interface pages direct from an UML class diagram

CodeGenie technical overview

- Reads XMI models from Rose, Together etc.
- Maps from UML 1.4
- Offers a model browse, metric and audit tool
- Generates code from a modelled architecture
- Uses Java archetypes for code generation
- Offers a Java/C++ Action Language library



- Generates C++, Java, J2EE, .NET, etc
- Written in Pure Java/Swing
- Used to generate itself

CodeGenie delivers

- Separation of the analysis models from the implementation environment
- Repeatable and defined software quality
- Shorter time-to-market
- Architectural and design flexibility
- Re-use of analysis and architecture models
- Comprehension of the problem

About Domain Solutions

Domain Solutions specialises in the application of the OMG's Model Driven Architecture (MDA) and Executable UML object-oriented software development methodologies and the development and exploitation of the CodeGenie code generation tools and techniques. The Company is also able to offer a broad base of software development and project management expertise. Domain Solutions is a dynamic software consultancy that operates across several market areas. Since 1995 we have specialised in delivering successful high quality object-oriented software solutions through requirements analysis, to design, architecture specification and programming.

For more information please contact:

NRT – Proventek Solutions AB
Enhagsvägen 8 B, 1tr
S-187 40 Täby

Phone: +46-8-792 25 80
Fax: +46-8-768 62 60

Email: office@nrt.se
Internet: <http://www.nrt.se>