CodeGenie

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(R)

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

Analyse your application

Don't just draw throw-away design diagrams in UML(R)! Analyse the problem & produce implementation-free diagrams that define the system!

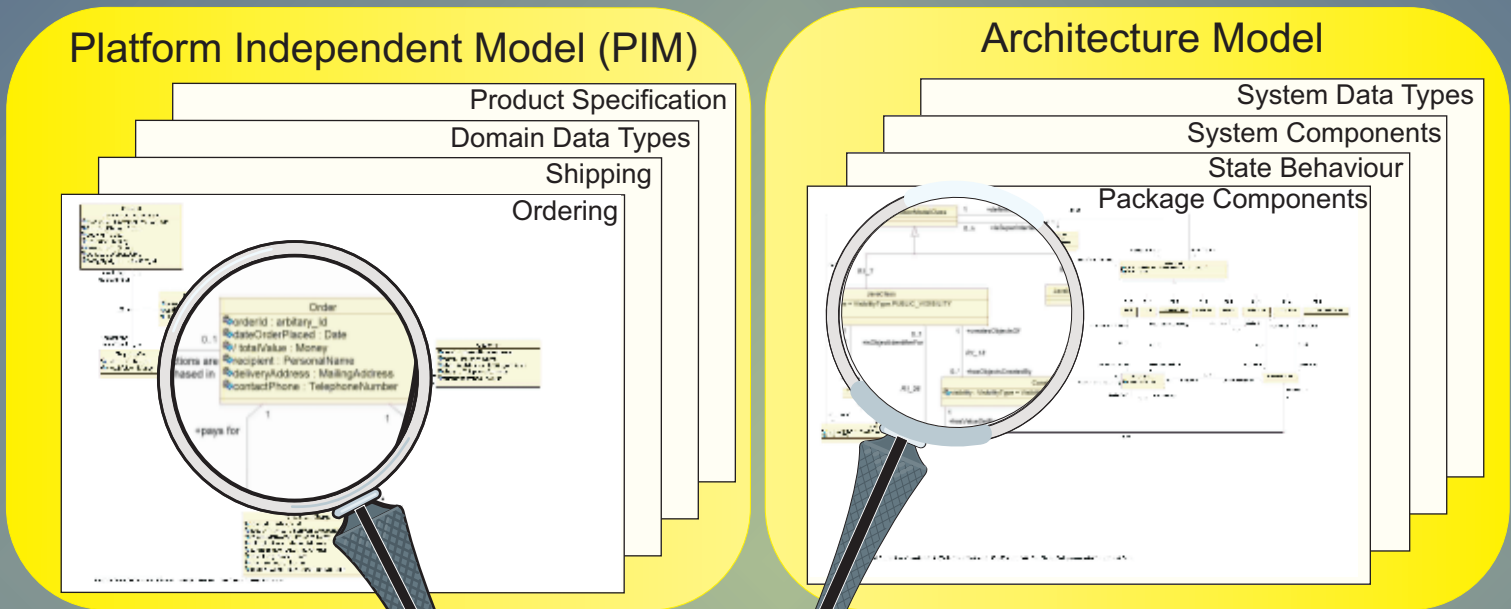MODEL DRIVEN ARCHITECTURE(R)

UNIFIED MODELING LANGUAGE
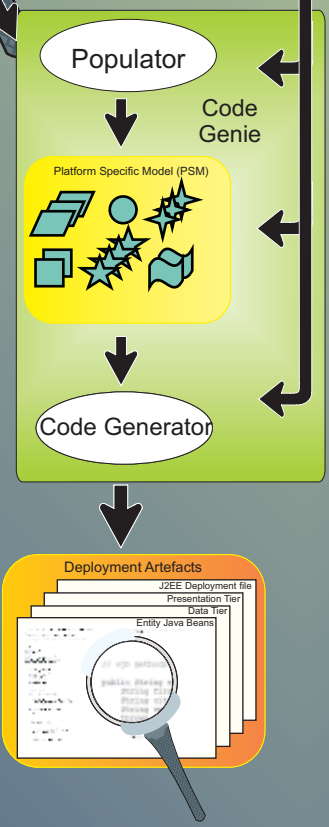
(C) Domain Solutions Ltd
Dec 04

What is a Platform Independent Model (PIM)?

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

Platform Independent Model (PIM)

Product Specification
Domain Data Types
Shipping
Ordering

Architecture Model

System Data Types
System Components
State Behaviour
Package Components

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 & can accommodate the client's software processes, target software architecture and software development tools.
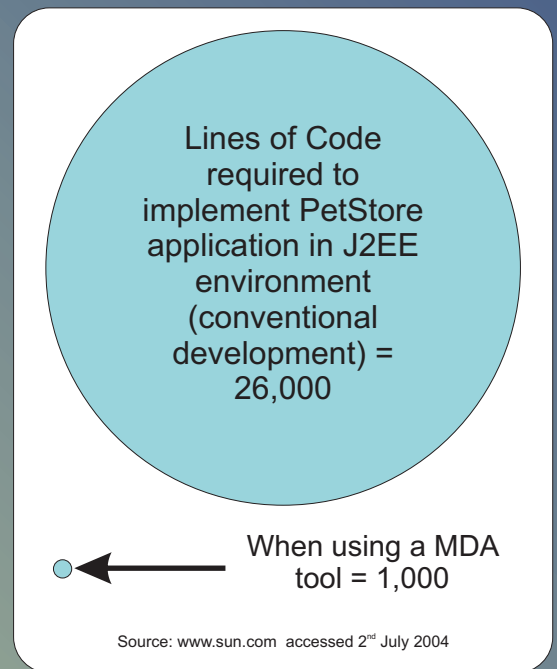
Populator

Code Genie

Platform Specific Model (PSM)

Code Generator

Deployment Artefacts
J2EE Deployment file
Presentation Tier
Data Tier
Entity Java Beans

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 graphic 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.

Lines of Code required to implement PetStore application in J2EE environment (conventional development) = 26,000

When using a MDA tool = 1,000

Source: www.sun.com accessed 2nd July 2004

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.

J2EE 2 or 3 tier architecture
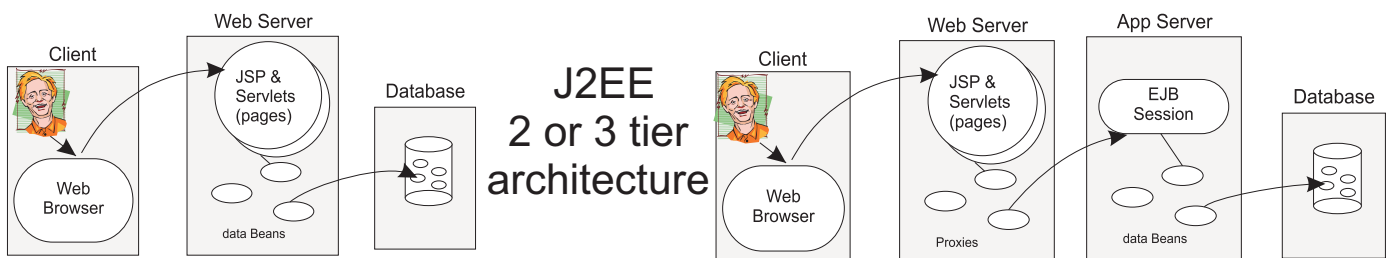
Source: http://research.sun.com/features/ace/AceHandout_OH2002.pdf accessed 7th June 2004

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[R] (XML Metadata Interchange[R]) 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.

(C) Domain Solutions Ltd

Case Study: AMS (BAE Systems) Radar Controller

o   Industrial Strength architecture built on repeating design patterns
o   Built on the OMG CORBA$^{(R)}$ platform offering object distribution
o   Generates around 80% of the system in IDL/C++
o   Several hundred thousand lines of code
o   Metrics show return on investment already gained on the first iteration!

Case Study: EuroControl ATC Course Management System

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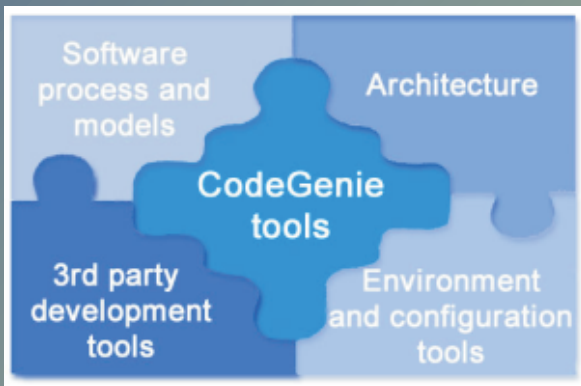
CodeGenie technical overview

o   Reads XMI models from Rose, Together etc.
o   Maps from UML.1.4
o   Offers a model browse, metrics and audit tool
o   Generates code from a modelled architecture
o   Uses Java archetypes for code generation
o   Offers a Java/C++Action Language library
o   Generates C++, Java, etc.
o   Written in Pure Java/Swing.
o   Used to generate itself!

CodeGenie delivers

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

Buckingham House
35 Graham Road, Malvern,
Worcestershire, WR14 2HU
England
http://www.ooagenerator.com
email: info@ooagenerator.com
Phone/fax +44 (0)1684 578875

VAT No: 661 3833 36
Registered in England No 3101732

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.