Model-Driven Architecture
Case Study

June 12th, 2003

Presented by:
David Bertrand
Director-Consulting, CGI Group
Table of content

- Ingredients Of The Approach
- Implementation Of The Approach
- Case Study
- Questions & Discussion
Ingredients Of The Approach
Main Ingredients

- Model-Driven Architecture (MDA)
- Rational Unified Process (RUP)
- Codagen Tools
Model-Driven Architecture

- **MDA - Specifications**

  - Vertical Markets
  - Business Domains
  - Abstracted Pervasive Services
  - Technology-Independent UML Environment
  - Middleware Environments Targeted For MDA
Model-Driven Architecture

- MDA - Models

- Business Function
  - Business Model
  - Platform Independent Model (PIM)
  - Platform Specific Model (PSM)

- Computation Independent

- Platform Independent
RUP Phases

- Inception
  - Vision
  - Risks
  - Scoping
  - Iterations Plan
- Elaboration
  - Requirements
  - Architecture
  - Built or Buy
- Construction
- Transition
RUP Workflows

- Development Workflows
  - Requirements Management
  - Analysis & Design
  - Implementation
  - Tests
  - Deployment

- Support Workflows
  - Project Management
  - Configuration & Change Management
  - Tools & Environment
**Model-Driven Architecture with Codagen**

The diagram illustrates the Model-Driven Architecture (MDA) approach using Codagen tools. It consists of:

- **Platform Independent Model (PIM)**
- **Platform Specific Model (PSM)**
- **Architecture and Validation Rules**

**Architectural Specification**:
- Layer
- Issue
- Property
- ...
- ...

**Architecture Implementation**:
- Layer
- Issue
- Template
- ...
- ...

The flow from PIM to PSM is facilitated by the architecture and validation rules.
Model

Model your applications in UML using your favorite modeling tool.

Extend

Extend your models with transformation markers and architecture-related data.

Transform

Transform your models into platform-specific source-code, documentation, SQL scripts, and much more!
Model-Driven Architecture with Codagen

Direct support (add-in) for:
- Rational Rose, TogetherSoft C/C, Microsoft Visio
- UML

Transform models into:
- Java, VisualBasic.NET, C#.NET, C++
- ASCII files (XML, ASP, JSP, HTML, etc.)

Extend models by:
- Defining and using UML Profiles

SPECIFY  MAP

IMPLEMENT  GENERATE
## Model-Driven Architecture with Codagen

<table>
<thead>
<tr>
<th><strong>Specify</strong></th>
<th>Used to define a UML Profile which is a custom definition of tagged values that will help refine the model.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Map</strong></td>
<td>Used to map platform-specific data to UML elements such as Packages, Classes, Attributes, etc.</td>
</tr>
<tr>
<td><strong>Implement</strong></td>
<td>Used to create and organize validation &amp; transformation templates.</td>
</tr>
<tr>
<td><strong>Generate</strong></td>
<td>Used to generate the code based on the mapping and the templates.</td>
</tr>
</tbody>
</table>
Model-Driven Architecture with Codagen

- **Specify**
  - UML Profile

- **Map**
  - Transformation Markers

- **Implement**
  - Validation & Transformation Templates

- **Generate**
  - Platform Specific Model (PSM)
    - Source Code
    - XMI Model
    - Documentation

Platform Independant Model (PIM)
Implementation Of The Approach
MDA & Inception/Elaboration Phases

Architecture
- Technological Architect
- Codagen Architect
- Rules & Templates

Requirements
- Business Architect/Analyst
- Modeling Tool
- Business Object Model (BOM)
- Use Cases

Analysis
- Business Analyst
- Modeling Tool
- Platform Independent Model (PIM)

Project/Configuration/Change Management
- PM Tool
- CM Tool
- Tracking Tool
Case Study
Context

Project Description

- Re-Create the systems top to bottom with new technologies.
- Technologies: Java/J2EE & C++/CORBA.
- Over 52,000 man/day project.
- Multiple projects in parallel.
- Very high specifications in term of security, integrity, availability and performance.
Context

- **Project’s Objectives**
  - **Productivity:** Development Cycle
  - **Reusability:** Architecture, Frameworks, Design Patterns, Components, etc.
  - **Integration:** Approach and Tools Kit
  - **Collaboration:** Between all the interveners of the development cycle
  - **Multi-Projects:** Approach and Tools Kit that support multiple projects realization in parallel.
Based on industry standards

- OMG’s Model-Driven Architecture approach.
- Rational Unified Process for the development process methodology.
- UML formalism.
- Enterprise tools recognised by the industry.
Gains of Productivity

- Based on reuse and code generation
- Define benchmarking process and evaluation criteria
- Code generators evaluation & benchmarking
- ROI calculation
- Code generator selection
- Measurement of productivity gains

Selected tools: Codagen
## Savings

<table>
<thead>
<tr>
<th></th>
<th>Without Architect</th>
<th>With Architect</th>
<th>Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Business Modeling</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capture business vision</td>
<td>20</td>
<td>20</td>
<td>0%</td>
</tr>
<tr>
<td>Define business rules</td>
<td>50</td>
<td>50</td>
<td>0%</td>
</tr>
<tr>
<td>Define scope</td>
<td>30</td>
<td>30</td>
<td>0%</td>
</tr>
<tr>
<td>Develop business use case model</td>
<td>75</td>
<td>75</td>
<td>0%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>175</td>
<td>175</td>
<td>0%</td>
</tr>
<tr>
<td><strong>Requirements Management</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Understand stakeholder needs</td>
<td>150</td>
<td>150</td>
<td>0%</td>
</tr>
<tr>
<td>Analyze the problem</td>
<td>200</td>
<td>200</td>
<td>0%</td>
</tr>
<tr>
<td>Define the system</td>
<td>250</td>
<td>250</td>
<td>0%</td>
</tr>
<tr>
<td>Manage the scope</td>
<td>100</td>
<td>100</td>
<td>0%</td>
</tr>
<tr>
<td>Refine the system definition</td>
<td>150</td>
<td>150</td>
<td>0%</td>
</tr>
<tr>
<td>Manage change</td>
<td>50</td>
<td>50</td>
<td>0%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>900</td>
<td>900</td>
<td>0%</td>
</tr>
<tr>
<td><strong>Analysis and Design</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Define a functional architecture of the system</td>
<td>200</td>
<td>200</td>
<td>0%</td>
</tr>
<tr>
<td>Detail the analysis packages</td>
<td>100</td>
<td>100</td>
<td>0%</td>
</tr>
<tr>
<td>Realize the use cases</td>
<td>150</td>
<td>150</td>
<td>0%</td>
</tr>
<tr>
<td>Detail the analysis classes</td>
<td>100</td>
<td>100</td>
<td>0%</td>
</tr>
<tr>
<td>Analyze the data needs</td>
<td>50</td>
<td>50</td>
<td>0%</td>
</tr>
<tr>
<td>Refine the system architecture</td>
<td>200</td>
<td>50</td>
<td>75%</td>
</tr>
<tr>
<td>Detail the subsystems and their interfaces</td>
<td>300</td>
<td>100</td>
<td>67%</td>
</tr>
<tr>
<td>Realize the interfaces of the subsystems</td>
<td>200</td>
<td>50</td>
<td>75%</td>
</tr>
<tr>
<td>Finalize the design classes</td>
<td>50</td>
<td>25</td>
<td>50%</td>
</tr>
<tr>
<td>Define the physical data model</td>
<td>100</td>
<td>100</td>
<td>0%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>1450</td>
<td>925</td>
<td>36%</td>
</tr>
<tr>
<td><strong>Implementation</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Define the implementation model</td>
<td>20</td>
<td>20</td>
<td>0%</td>
</tr>
<tr>
<td>Define the integration plan</td>
<td>20</td>
<td>20</td>
<td>0%</td>
</tr>
<tr>
<td>Carry out template implementation and code generation</td>
<td>0</td>
<td>50</td>
<td>0%</td>
</tr>
<tr>
<td>Integrate each subsystem</td>
<td>50</td>
<td>50</td>
<td>50%</td>
</tr>
<tr>
<td>Develop specific code and execute unit test</td>
<td>500</td>
<td>150</td>
<td>63%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>540</td>
<td>290</td>
<td>46%</td>
</tr>
<tr>
<td><strong>Test and QA</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Develop a plan for functional tests</td>
<td>30</td>
<td>30</td>
<td>0%</td>
</tr>
<tr>
<td>Design functional tests</td>
<td>150</td>
<td>150</td>
<td>0%</td>
</tr>
<tr>
<td>Execute functional tests</td>
<td>350</td>
<td>200</td>
<td>43%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>530</td>
<td>380</td>
<td>28%</td>
</tr>
</tbody>
</table>

**TOTAL**                  | 3595              | 2670           | 26%     |
Savings

Process with MDA

Process without MDA

SAVINGS

Effort / $$

Time

Requirements | Analysis/Design | Implementation | Test | Deploy | Maintenance
Benefits

- **Concrete gains of productivity**
  - Decrease time to market
  - Cost savings

- **Major reduction of the development team**
  - Few seniors (architects) define, implement and maintain the architecture
  - UML Modelers define and maintain the business
  - Few junior developers code the specific (business rules and validation)
  - Don’t need anymore an army of senior or specialist developers

- **Homogeneous & uniform architecture**

- **Simple development process**
  - Very well defined activities and roles
  - Very well defined artifacts and deliverables to produce