Case Study:

**Setting up a Domain Specific Tooling**

Stéphane Lacrampe - CEO  :  stephane.lacrampe@obeo.fr
Cédric Brun - Eclipse Modeling expert  :  cedric.brun@obeo.fr
Agenda

- Who is Obeo?
- Some background on Unédic and Eclipse
- Building a Domain Specific Model
- Building a Domain Specific Modeler
- Code generation
- Packaging and deployment
- Conclusion
Obeo : Model Driven Company

- OSS Vendor since 2005
  - Model Driven experts
  - 20 employees by end 2007
  - Nantes, Paris by end 2007
  - Self funded and profitable

- Our customers and partners
Our products and services

Our MDSD offer:
- “Designing Software Factories based on Eclipse technology”
- Consulting on Eclipse, EMF, GMF, M2M, M2T ...
- Support, training

Our ADM offer:
- “Model Driven Migration Factories, software cartography”
- Available for JEE, VB, ASP, C, C++, Ada, Forte, SQL, Progress...

Our products:
- Acceleo: http://www.acceleo.org
- Acceleo Pro Traceability, Acceleo Pro DSM
- Agility
Obeo – Eclipse foundation member
- 7 Eclipse committers
- Eclipse Modeling Project
  - EMFT: EMF Compare (leader: Cedric Brun)
  - M2M: QVT Relation (leader: Quentin Glineur), ATL
  - Modisco
- Eclipse SOA Tool Platform
  - SCA Composite Editor – IP review in process

OW2 Consortium (ex-ObjectWeb)
- Acceleo
- OW2 Board member

Collaborative project and R&D
- Topcased member (http://www.topcased.org)
- Papyrus
- Clusters System@Tic
- Research (Lina, Inria, Imag, ...)

Collaborative project and R&D
Agenda

- Who is Obeo?
- Some background on Unédic and Eclipse
- Building a Domain Specific Model
- Building a Domain Specific Modeler
- Code generation
- Packaging and deployment
- Conclusion
• Creation: 31st, December 1958

• French Administration, 15,000 employees

• Missions
  – Job seekers - social care
  – Assedic (unemployed people administration) coordination

• IT department
  – 1,500 employees
  – Outsourcing
Business applications are built on a 4 layers architecture - JEE

- **Layer P (Client/Server)**: Web browser + Java Applet, Web browser, XML rendering
- **Layer N (Context and state machine)**: Folder portal (Web browser + Java Applet), Web portal (Web browser), Other customer, XML portal
- **Layer L (Business Services)**: Business Service A, Business Service B
- **Layer D (Data Persistence)**: Component, Database
Eclipse tooling

- Eclipse: an early choice
  - Eclipse as a Java IDE
  - Modeler: Rational XDE
  - 2004: design of 2 Eclipse based tools
    - Analysis and Design
    - GEF + Ecore + Jet

OUDINI V2 Tools

- Entity Creation
- Code generation

Wizards

Screen mock-up
- Screens
- Folder tree

XML Editor
- Functions
- User states
- Data structure
- Presentation objects

Java code editor
- Functions
- User states

Unedic Plug-In

ECLIPSE Platform

Referential
- XML presentation entities
- XML Navigation entities
- Java code

Clear Case
Tooling state early 2006

The software factory is started but models are still not productive
2006 – 2007: towards EMF, GMF and Acceleo

- **Architecture evolution**
  - JSF/Ajax
  - Tooling improvement

- **From UML to a DSM approach**
  - Language closer to the Unédic logical/functional concepts
  - Based on Eclipse standards: EMF, GEF, GMF

- **Generative approach with Acceleo Pro**
  - Automate projection between functional models and Unedic JEE Framework
  - MDSD and Traceability

- **Unédic collaborates with Obeo on their Software Factory**
The whole picture

- Two phases project
  - Prototype
  - Industrial version

- Main steps
  - DSM Design
  - EMF-GMF implementation
  - Extended features
  - Code generation
  - Packaging and deployment
● Who is Obeo?

● Some background on Unédic and Eclipse

● Building a Domain Specific Model

● Building a Domain Specific Modeler

● Code generation

● Packaging and deployment

● Conclusion
• How to design a Domain Model?
  - Find out Business concepts
  - Cut them into logical meta-models

• Who
  - Modeling tool expert
  - Business and methodology expert
Context meta-model

Un Objet de Contexte peut faire référence à d'autres Objets de contexte ou objets de transferts provenant du modèle L.

Selon la stratégie de mapping sur le N, une Structure peut faire référence à des Objets de contexte ou des Objets de Transferts.
Domain Model Implementation

- Ecore implementation
  - Topcased Ecore editor
  - EMF standard generation
  - Icons

- Difficulties
  - A good meta-model?
  - Model hierarchy
  - GMF and tooling impacts
  - Not all Business rules modeled

- A key task!

- Around 100 meta-elements
Agenda

- Who is Obeo?
- Some background on Unédic and Eclipse
- Building a Domain Specific Model
- Building a Domain Specific Modeler
- Code generation
- Packaging and deployment
- Conclusion
Designing a DSM

- **Choices to make:**
  - How to graphically represent those concepts for the end-user?
  - How many diagram types?
  - Diagram root and breakdown

- **Choices based on:**
  - Domain
  - Know-how on GMF tooling (GMF V1)

- **3 diagrams:**
  - Navigation: screen dynamic behaviour
  - Context: data used and shared by screens
  - Types: complex data structure description
Implementing a DSM with GMF

- Steps to build a Domain Specific Modeler with GMF
  - Set-up the GMF models (graphical, tooling, mapping)
  - Generate Java code from gmfgens

- Extending GMF
  - Using GMF extension points when possible
  - Modifying generated code

- Extensions
  - Properties edition on double click
  - Creation wizards
  - Diagram navigation and shortcuts
  - Diagram synchronization
  - Model refactoring, Model validation, Clearcase integration
Implementing a DSM with GMF

Demo
• Who is Obeo?

• Some background on Unédic and Eclipse

• Building a Domain Specific Model

• Building a Domain Specific Modeler

• Code generation

• Packaging and deployment

• Conclusion
• Acceleo: http://www.acceleo.org
  - Open source (EPL) code generator based on Eclipse and EMF
  - Easy learning curve, end user features
  - Incremental generation, performance
  - Template editors, debugging
  - Plug-in packaging of generators
  - Generation module repository (JEE, C#, Php, Python...)

• Acceleo Pro Traceability:
  - Model-code-Model navigation and round-trip
  - Generation preview, de-synchronization detection
  - Partial regeneration, generated code modification
How to design a code generator from a Domain Model
- Have a prototype of your generation target
- Identify patterns and things that can be automated
- Make sure that you've got the required information in models

Two generators
- Application code: Java, JSP and XML code generator
- Documentation: HTML

Around 70 templates
Example of a Unedic application
Example of a Unedic application
Example of a Unedic application
• Who is Obeo?

• Some background on Unédic and Eclipse

• Building a Domain Specific Model

• Building a Domain Specific Modeler

• Code generation

• Packaging and deployment

• Conclusion
Packaging and deployment

- Packaging
  - 20 plug-ins (Eclipse 3.2)
  - Build issues (GMF/JDK 1.4/Windows)

- Deployment
  - Tools delivered to a third party company
  - Eclipse bundle delivery (Eclipse, Modeling, WTP...)
  - Regular updates through Eclipse update sites
  - User training
Agenda

- Who is Obeo?
- Some background on Unédic and Eclipse
- Building a Domain Specific Model
- Building a Domain Specific Modeler
- Code generation
- Packaging and deployment
- Conclusion
Feedback on the project

- **Time and duration**
  - Prototyping: 35 m.d
  - Industrial version: 95 m.d
  - Duration: 5 months

- **Unédic**
  - Specification, tests and validation, user documentation
  - Code generation

- **Obeo**
  - Meta-modeling and **Eclipse tooling**
  - EMF-GMF implementation
  - Acceleo training and coaching
  - Collaboration using Trac system + Mylyn
• Customer results
  - Deployment of the tooling on an **industrial project** with Unilog (LogicaCmg) - 5000 m.d
  - 15 persons in **parallel** using the software factory since June 2006
  - Huge **productivity gains**, easiest maintenance
  - Migration of existing projects
  - Standard and integrated tooling made on a **tight schedule and budget**

• Difficulties
  - Meta-models **evolution** impacts
  - Testing the GUI
  - Clearcase integration
  - Numerous models management with shortcuts (coherency, validation)
  - GMF2 migration ?