



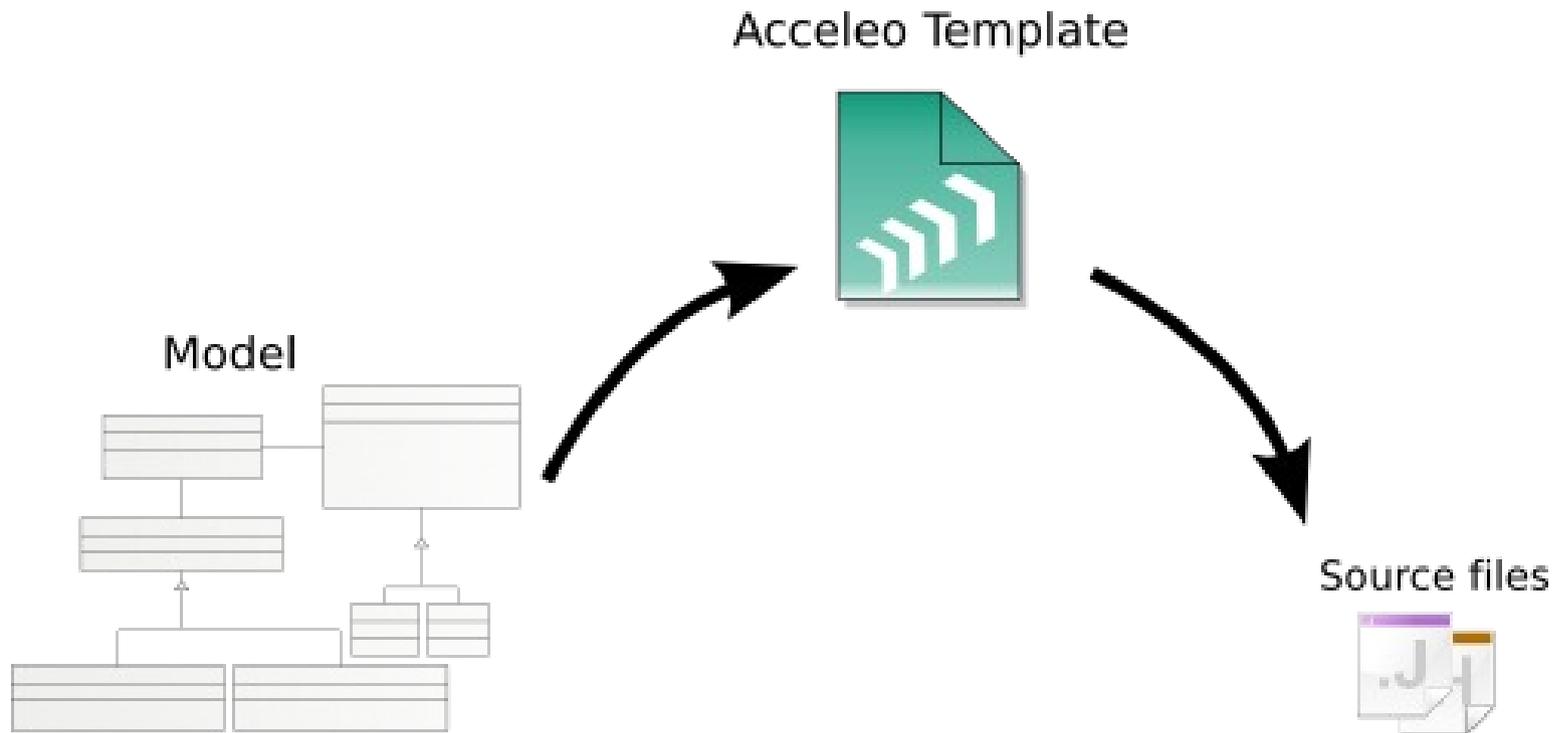
*Un générateur open source pour des développements
dirigés par les modèles*

Cédric Brun - Obeo

- Introduction
 - Un générateur Open Source ?
 - Quel intérêt ?
- Le développement piloté par les modèles
 - Principes
 - Apports
 - Pièges
- Une application J2EE avec Acceleo
 - Modélisation UML
 - Templates de génération
 - Pragmatisme
- Conclusion

- Introduction
 - **Un générateur Open Source ?**
 - Quel intérêt ?
- Le développement piloté par les modèles
- Une application J2EE avec Acceleo
- Conclusion

- Acceleo génère du texte depuis un modèle abstrait



modèle UML, DSL ...

module de génération : un ensemble de templates ciblant une technologie



```
WebLog_fr.uml  uml2toXhtml.mt X
<%
metamodel http://www.eclipse.org/uml2/2.0.0/UML
%>

<%script type="uml.Class" name="uml2toXhtml" file="<%name%>.html"%>
<html>
  <head/>
  <body>
    <h1>Class Description</h1>
    <p>Name of class : <%name%></p>
    <p>Comment : <%ownedComment.body%>

    <h1>Attributes</h1>
    <%if (attribute.nSize() == 0){%>
    <p>No attributes.</p>
    <%}else{%>
    <ul>
      <%for (attribute){%>
      <li><%name%> : <%type.name%></li>
      <%}%>
    </ul>
    <%}%>

  </body>
</html>
```

```
<html>
  <head/>
  <body>
    <h1>Class Description</h1>
    <p>Name of class : Utilisateur</p>
    <p>Comment :

    <h1>Attributes</h1>
    <ul>
      <li>email : String</li>
      <li>prenom : String</li>
      <li>nom : String</li>
      <li>login : String</li>
      <li>motDePasse : String</li>
    </ul>
  </body>
</html>
```

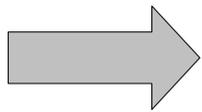


- Historique
 - v1.0.0 : Mars 2006
 - v1.1.0 : Septembre 2006
 - v1.2.0 : Janvier 2007
- Logiciel libre (sous licence EPL)
 - communauté, subversion, bug tracking, mailling lists
- Initialement développé par la société Obeo
 - spécialistes de la mise en oeuvre d'une démarche dirigée par les modèles
 - éditeur de solutions pour l'industrialisation du développement
 - conseil, formation

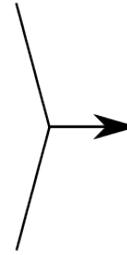
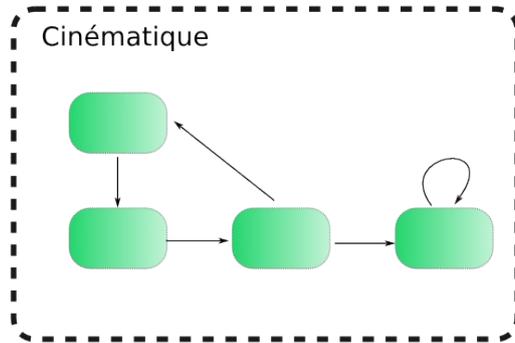


- Introduction
 - Un générateur Open Source ?
 - **Quel intérêt ?**
- Le développement piloté par les modèles
- Une application J2EE avec Acceleo
- Conclusion

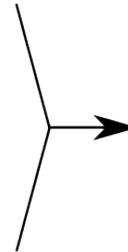
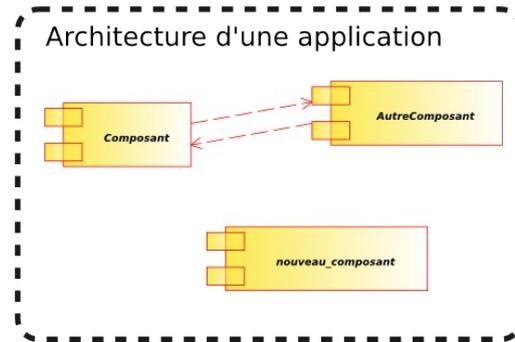
- L'utilisation des serveurs d'applications implique:
 - un découpage important du système
 - une dé-corrélation des éléments du système
 - de multiples technologies spécialisées pour chaque préoccupation



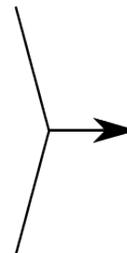
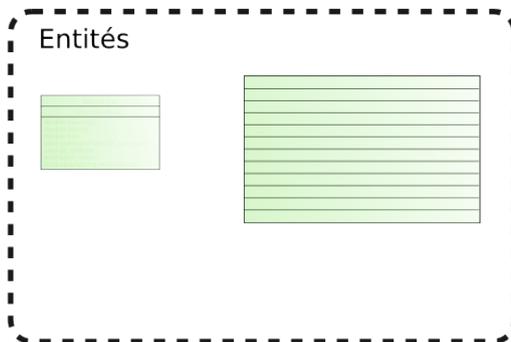
Multiplication du code « enveloppe »
Maîtrise délicate de l'ensemble



Feuilles JSP
Actions struts
StrutsConfig.xml



Interface du service
EJB Remote et Local



Objets de transferts



Fabrique de DAO
Interface de DAO
Implémentation de DAO
Mapping Hibernate

- Introduction
- Le développement piloté par les modèles
 - **Principes**
 - Apports
 - Pièges
- Une application J2EE avec Acceleo
- Conclusion

- Le modèle est positionné au centre du développement
 - le modèle devient productif
 - synchronisation modèle / code
- Le développement du logiciel passe par l'utilisation d'une fabrique
- Différents modèles pour différentes préoccupations
 - modèle abstrait proche du métier
 - modèle technique proche de la plateforme

Modélisation

Fonctionnel

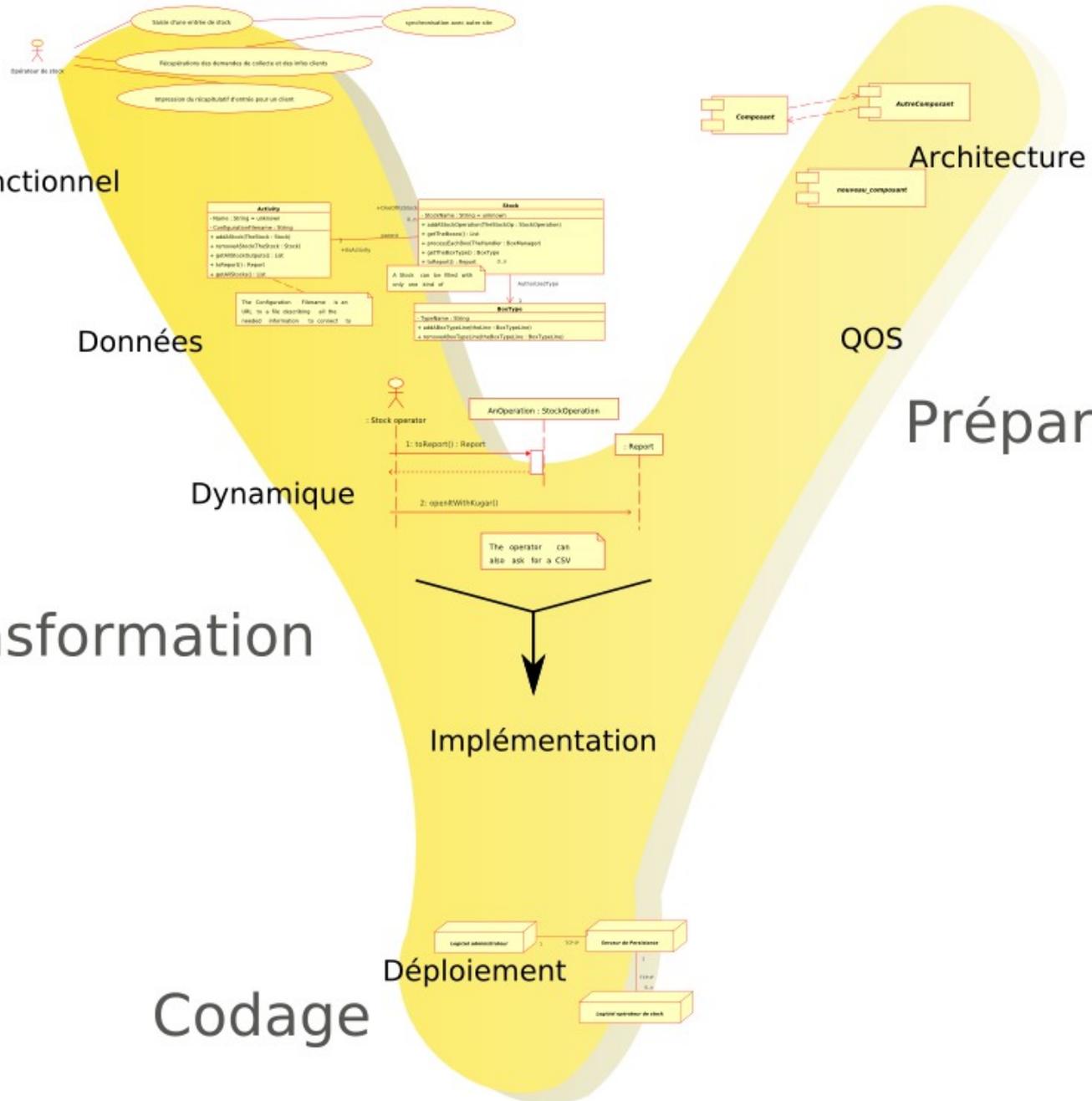
Données

Dynamique

Transformation

Implémentation

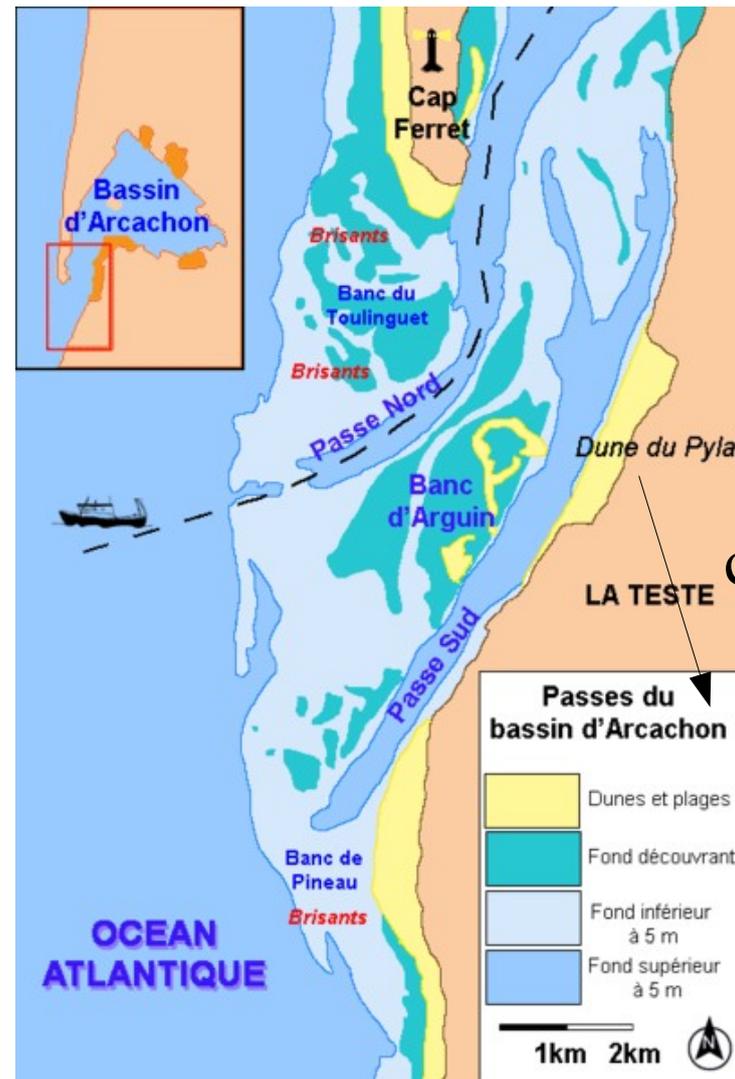
Codage



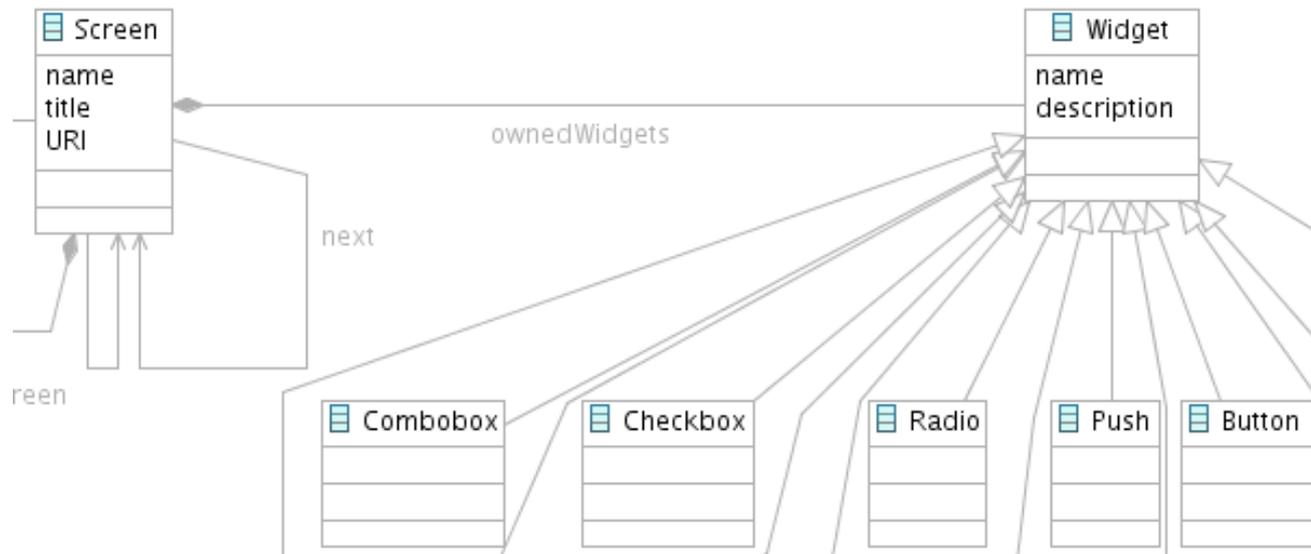
- **Modèle** : Représentation d'un système complexe
- **Méta-modèle** : Langage permettant de formaliser un modèle



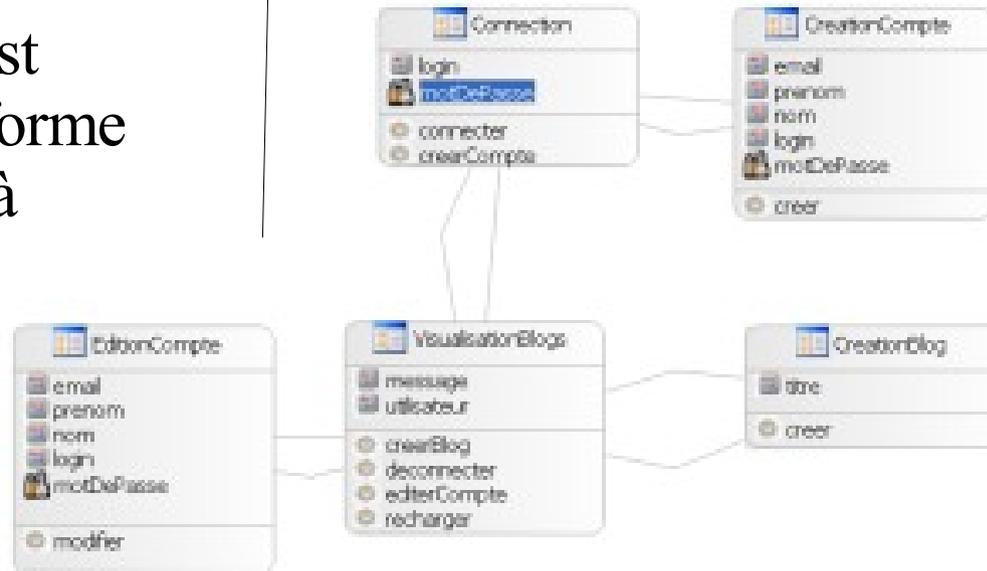
← Représente



est conforme à



est conforme à



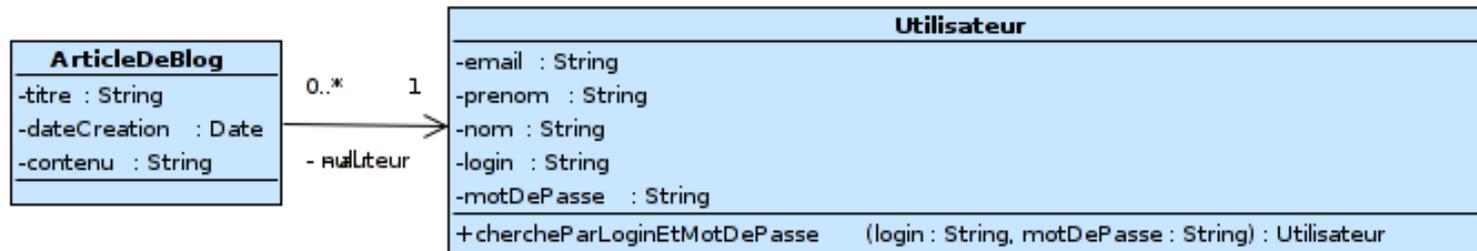
- Introduction
- Le développement piloté par les modèles
 - Principes
 - **Apports**
 - Pièges
- Une application J2EE avec Acceleo
- Conclusion

- Génération du code depuis le modèle
 - Gain sur les temps de développement
 - Mise en place de « bonnes pratiques » d'architecture
 - Nommage, normes qualités
- Indépendance vis à vis des technologies
- Maîtrise du système développé

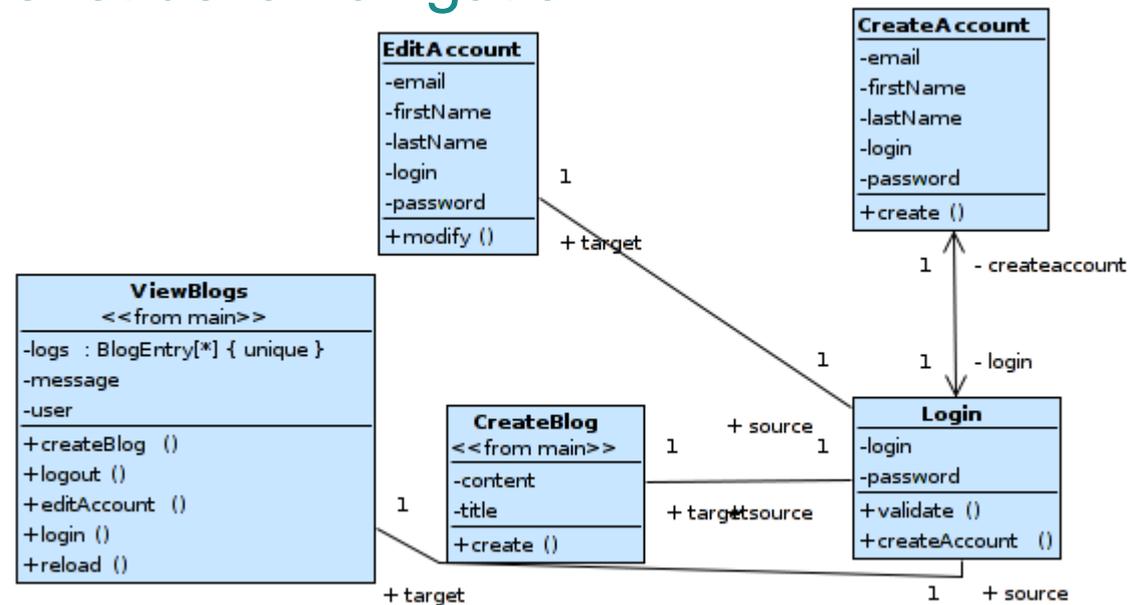
- Sur-modélisation
- Contraintes trop fortes sur le développement
- Démarche peu itérative
- Problème d'outillage
 - modeleurs
 - générateurs de code

- Introduction
- Le développement piloté par les modèles
- Une application J2EE avec Acceleo
 - **Modélisation UML**
 - Templates de génération
 - Pragmatisme
- Conclusion

- Modélisation :
 - des entités métiers



- des écrans et de la navigation



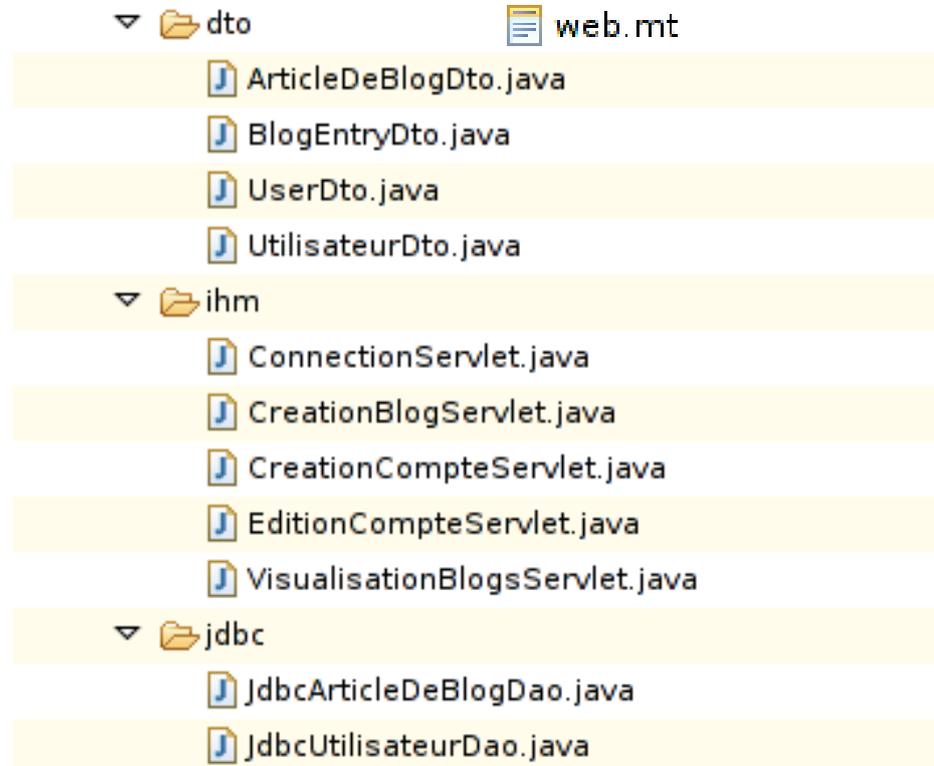
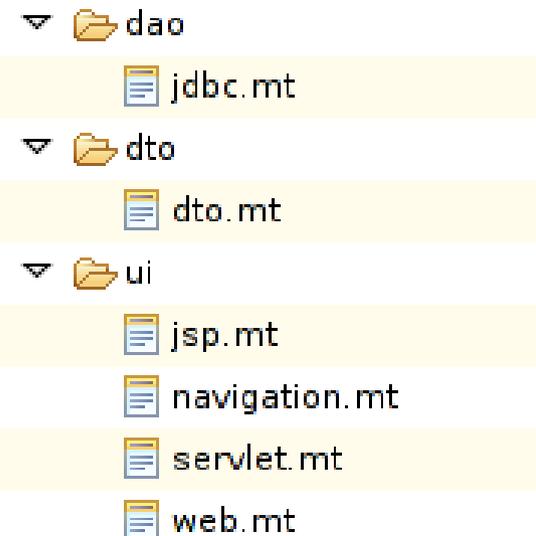
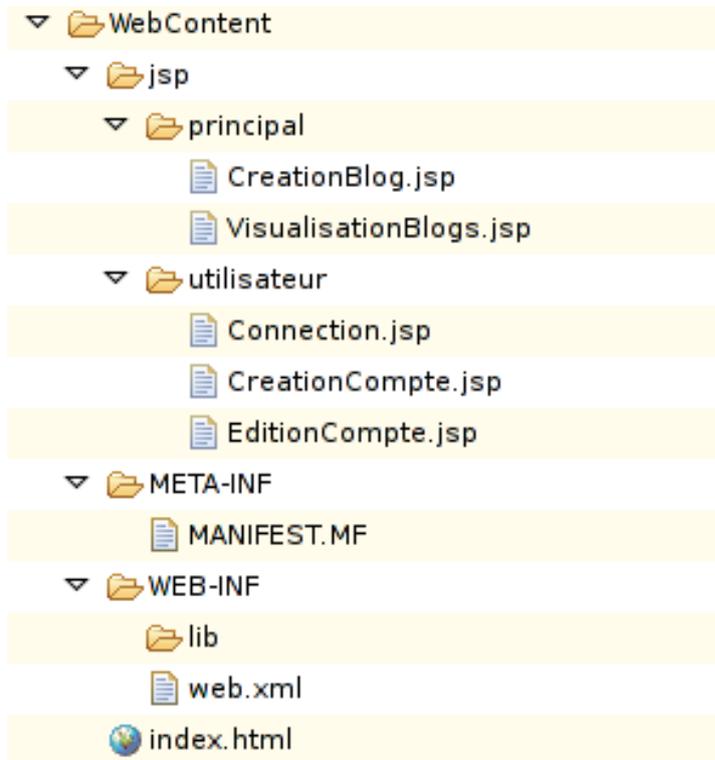


- Utilisation de stéréotypes
 - « Screen »
 - « Entity »
- Utilisation de tagged values
- Profiles UML
- D'autres solutions existent
 - Les méta-modèles personnalisés (Domain Specific Language)

- Introduction
- Le développement piloté par les modèles
- Une application J2EE avec Acceleo
 - Modélisation UML
 - **Templates de génération**
 - Pragmatisme
- Conclusion



- Module d'exemple :
 - DAO avec implémentation JDBC
 - DTO pojo
 - JSP avec Servlets





```
<%script type="Class" name="generate" file="<%fullFilePath%>"%>
<!-- DAO implementation generation --%>
package <%jdbcPackage%>;

import java.sql.Connection;
import java.sql.Date;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

import org.acceleo.fwk.dao.DaoException;
import org.acceleo.fwk.dao.jdbc.JdbcConnectionUtils;
import org.acceleo.sample.dto.<%name%>Dto;

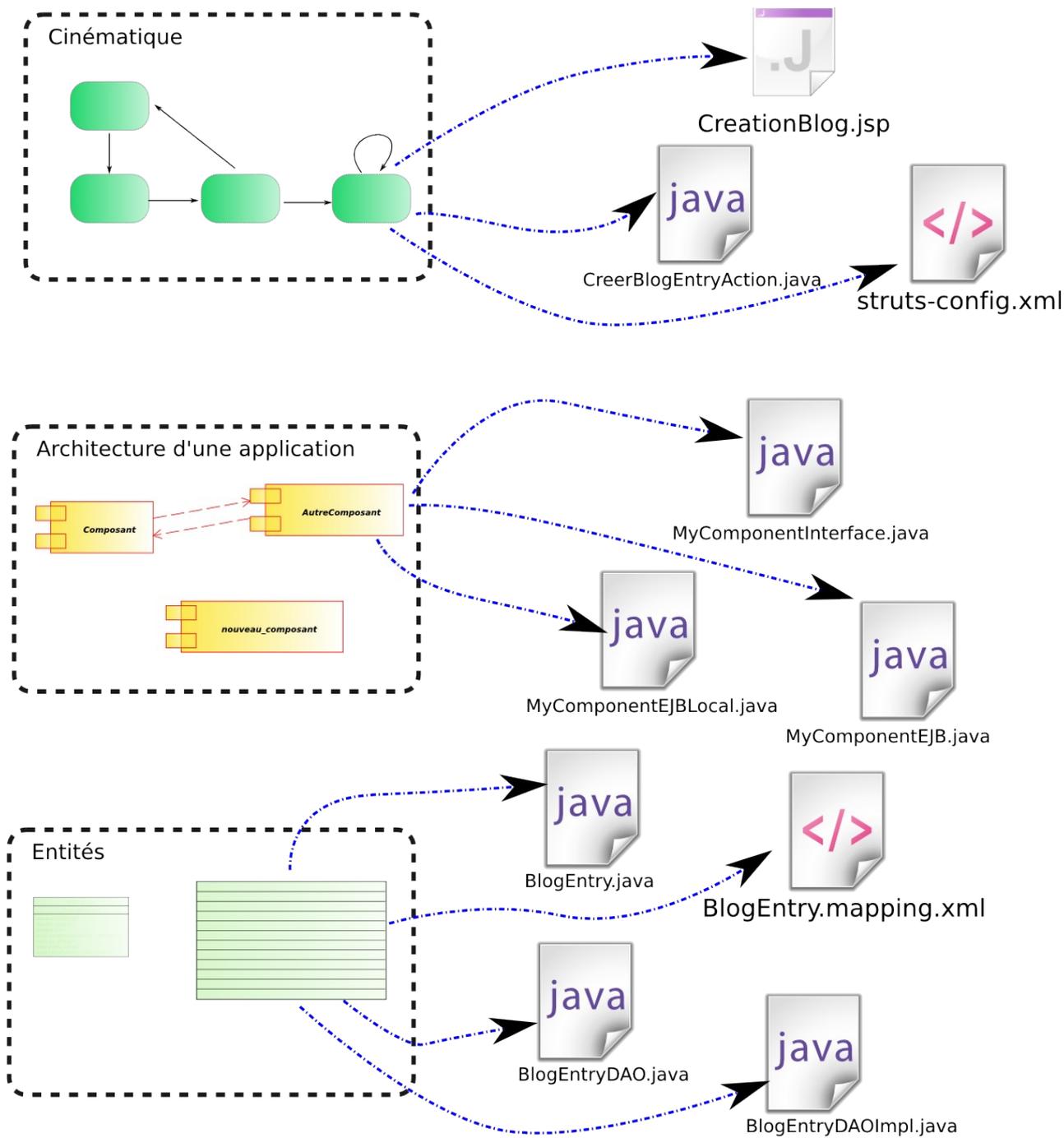
/**
 * This class provides the data access layer to the <%name%> entity class.
 * This is the JDBC implementation.
 */
public class Jdbc<%name%>Dao {

<!-- Generates static declaration for database fields --%>
    protected final static String TABLE_NAME = "T_<%name%>";

<%for (attribute){%>
    protected final static String <%name.toUpperCase()%> = "<%name%>";
<%}%>

    protected static final String CREATE_QUERY =
        "INSERT INTO " + TABLE_NAME +
        " (<%attribute.name.sep(", ")%>) VALUES ( <%for (attribute){%>? <%}%>)" ;
```

Templates de génération

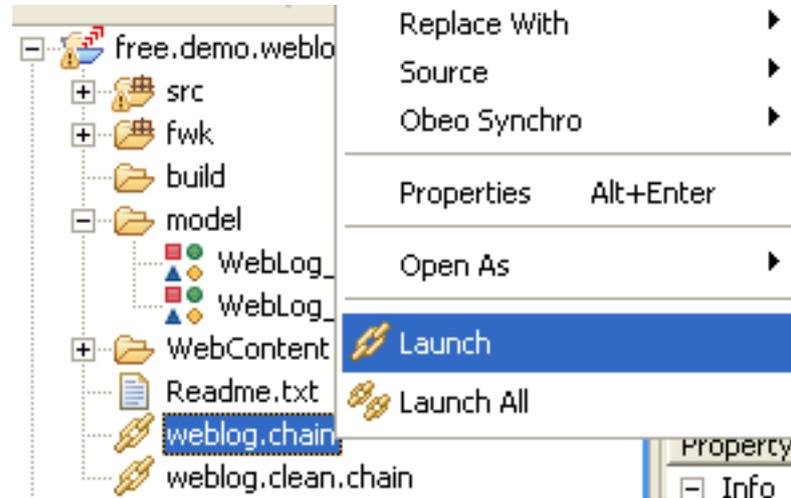


Confidentiel - Ce document est la propriété de Obeco et ne saurait être communiqué à un tiers sans son autorisation.

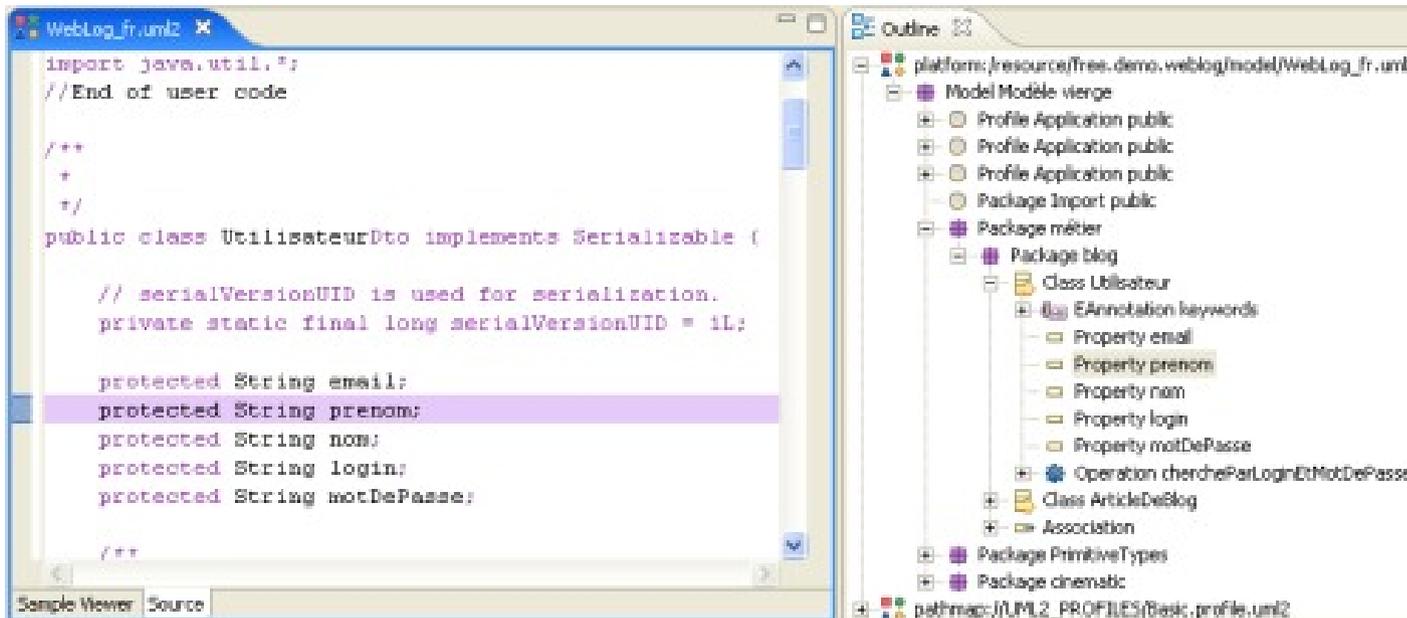
- Introduction
- Le développement piloté par les modèles
- Une application J2EE avec Acceleo
 - Modélisation UML
 - Templates de génération
 - **Pragmatisme**
- Conclusion

- 90% des choses sont simples
 - syntaxe simple pour les templates
 - du Java pour les 10% restants
- Éviter la sur-modélisation
 - 1 élément modélisé => beaucoup de code généré
- Faciliter les itérations
 - génération incrémentale
 - pré-visualisation du code généré
- S'intégrer au processus de développement
 - s'intégrer à la gestion de versions
 - s'intégrer à l'environnement de développement

- Fichier de lancement



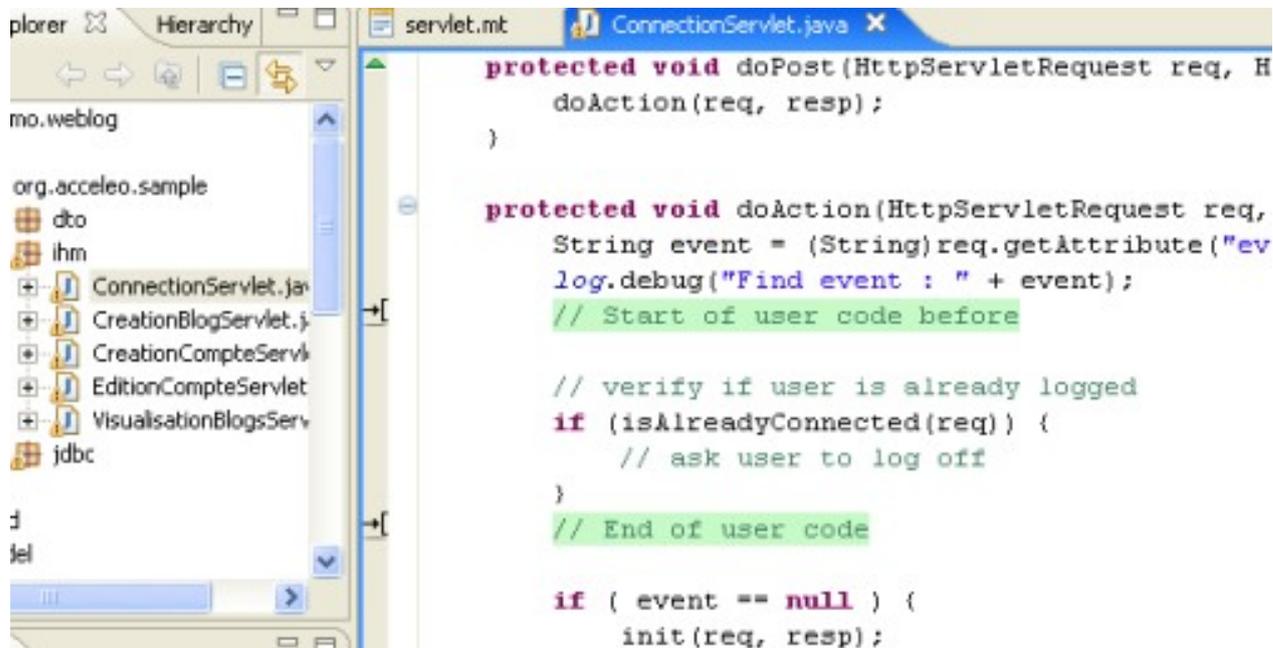
- Éditeur réflexif avec pré-visualisation instantanée



- Syntaxe simple spécialisée pour la génération

```
protected void doAction(HttpServletRequest req,
    String event = (String)req.getAttribute("ev
    log.debug("Find event : " + event);
    // <%startUserCode%> before
    // TODO Write here the code to execute befo
    // <%endUserCode%>

if ( event == null ) {
    init(req, resp);
}
```



- Éditeurs complets
 - assistance à la création des templates

```

<%script type="Class" name="generate" file=
<!-- DTO class generation-->
package <%dtoPackage%>;

//<%startUserCode%>
import java.io.Serializable;
import java.util.*;
//<%end UserCode%>

/**
 * Unresolved comment <%comment%>
 */
public class <%name%>Dto<%extensions%> impl
    
```

```

<%script type="uml2.Class" name="generate" file="<%fullFilePa
<!-- DTO class gener
package <%dtoPackage

//<%beginUserCode%>
import java.io.Seria
import java.util.*;
//<%endUserCode%>

/**
 * <%ownedComment.body%>
 */
    
```

- uml2.Class
- uml2.Classifier
- uml2.ClassifierTemplateParameter
- uml2.Clause
- uml2.ClearAssociationAction
- uml2.ClearStructuralFeatureAction
- uml2.ClearVariableAction

- Standards

- UML 1.3, 1.4 et 2.0
- XMI
- framework EMF



- Ouverture

- Multi-technologies cibles
- Appel de code Java pendant la génération



- Introduction
- Le développement piloté par les modèles
- Une application J2EE avec Acceleo
- **Conclusion**



- MDA + J2EE
 - architecture de référence
 - implémentation « par défaut »
 - modèle exploité => « code enveloppe »
 - synchronisation modèle/code
 - formation
- À garder à l'esprit
 - outillage
 - pragmatismes
 - non-intrusion dans le processus de développement

Merci pour votre attention !

- *Références*

- *Site du projet Acceleo : www.acceleo.org*
- *Obeo : www.obeo.fr*