



*Ministère de l'enseignement supérieur
et de la recherche scientifique*



Université de M'sila
Faculté des Mathématiques et de l'Informatique
Département de l'informatique

Mémoire de fin d'études
Pour l'obtention de Diplôme de **Master** en Informatique

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Informatique fondamentale

Thème

Transformation des diagrammes de classes
vers
des documents XML
suivant la démarche MDA

Réalisé par :
Bey Abdelaali

Dirigé par :
Mr. Mokhtari Rabah

2010 /2011

Table des matières

CHAPITRE II: Le rôle de XML dans la transformation des modèles et ATOM3	
2.1. Introduction.....	23
2.2. Modèles et langages de modélisation.....	23
2.2.1. UML (Unified Modeling Language).....	23
2.3. Langages de métamodélisation MOF 2.0 (Meta Object Facility).....	25
2.4. Format d'échange normalisé : XMI.....	27
2.4.1. Extensible Markup Language (XML).....	28
2.4.2. Format d'un document XML.....	28
1.2.4.1. XML Schema.....	29
1.2.4.2. Définition de Type de Document.....	29
2.5. L'utilisation des standards pour la transformation.....	30
2.6. Transformation des modèles du MDA.....	30
2.6.1. Type de Transformation de modèle.....	31
2.6.1.1. De PIM vers PIM.....	31
2.6.1.2. De PIM vers PSM.....	32
2.6.1.3. De PSM vers PSM.....	32
2.6.1.4. De PSM vers PIM.....	32
2.6.1.5. La transformation par annotation ou marquage.....	33
2.7. Classification des approches de transformation.....	34
2.7.1. Transformations de type Modèle vers code.....	34
2.7.2. Transformations de type modèle vers modèle.....	35
2.7.1.1. Structure d'une transformation.....	35
2.8. Différentes approches.....	37
2.8.1. Approches par manipulation directe.....	37
2.8.2. Approches relationnelles.....	37
2.8.3. Approches basées sur les transformations de graphes.....	37
2.8.4. Approches basées sur la structure.....	38
2.8.5. Approches hybrides.....	38
2.9. Grammaires de graphes.....	38
2.10. Outils de transformation de graphes.....	38
2.11. AToM3 : Présentation générale.....	39
2.12. Conclusion.....	40
 CHAPITRE III: Contribution et teste	
3.1. Introduction.....	42
3.2. Diagramme de classes en XML.....	42
3.3. L'approche proposée.....	43
3.3.1. Meta-modèles et la génération de l'outil de transformation.....	44
3.3.1.1. Meta-modèles des diagrammes de classes.....	44
3.3.1.2. Génération de l'outil des diagrammes de classes.....	45
3.2. Grammaire de graphe et génération de document XML.....	45

Table des matières

3.2.1 .Grammaire de graphe.....	45
3.2.1.1. Action initiale.....	45
3.2.1.2. Action finale.....	46
3.2.2. Règles de transformations.....	47
3.2.2.1. Première règle SelectfirstClasse.....	47
3.2.2.2. Deuxième règle select1heritage.....	48
3.2.2.3. Troisième règle SelectClasseMeme.....	50
3.2.2.4. Quatrième règles Classe2Classe.....	51
3.2.2.5. Cinquième règle Classe2ClasseH.....	53
3.2.2.6. Sixième règle Classe2Meme.....	55
3.2.2.7. Septième règle IsoliteClasse.....	56
3.3. Création d'un bouton d'exécution.....	57
3.4. Exemple.....	58
3.4.1. Fonctionnement pédagogique de l'université.....	58
3.5. Transformation du diagramme de classes XML.....	59
3.6. Conclusion.....	61
Conclusion GENERALE.....	63
BIBLIOGRAPHIE.....	
Figure 3.1: Diagramme de classe générique document XML.....	42
Figure 3.2: XML schema pour les diagrammes de classes.....	43
Figure 3.3 : Le Meta-Model pour les diagrammes de classes.....	44
Figure 3.4 : Outil générique pour le diagramme de classes.....	45
Figure 3.5 Exemple de code d'Action initiale.....	46
Figure 3.6 Exemple de code d'Action finale.....	46
Figure 3.7: LHS et RHS de SelectfirstClasse.....	47
Figure 3.8 : Condition de la règle SelectfirstClasse.....	48
Figure 3.9 : Action de la règle SelectfirstClasse.....	48
Figure 3.10: LHS et RHS de Select1heritage.....	49
Figure 3.11 : Condition de la règle Select1heritage.....	49
Figure 3.12 : Action de la règle Select1heritage.....	50
Figure 3.13: LHS et RHS de SelectClasseMeme.....	50
Figure 3.14 : Condition de la règle SelectClasseMeme.....	51
Figure 3.15 : Action de la règle SelectClasseMeme.....	51
Figure 3.16: LHS et RHS de Classe2Classe.....	52
Figure 3.17 : Condition de la règle Classe2Classe.....	52
Figure 3.18 : Action de la règle Classe2Classe.....	53
Figure 3.19: LHS et RHS de Classe2ClasseH.....	53
Figure 3.20 : Condition de la règle Select1heritage.....	54
Figure 3.21 : Action de la règle Classe2ClasseH.....	54
Figure 3.22: LHS et RHS de SelectClasseMeme.....	55
Figure 3.23 : Condition de la règle SelectClasseMeme.....	55

INTRODUCTION GÉNÉRALE

L'OMG (Object Management Group) a défini sa nouvelle orientation pour résoudre les problèmes actuels de construction et d'évolution de systèmes d'information auxquels font face les industriels. Cette orientation se nomme MDA (Model Driven Architecture) et se démarque très nettement de l'ancienne (OMA : Object Management Architecture), en se positionnant clairement au niveau supérieur d'abstraction et en mettant le projecteur non plus sur les approches à objets mais sur les approches à modèles.

Le but est de se concentrer sur l'élaboration de modèles de plus haut niveau d'abstraction et de favoriser les approches transformationnelles paramétriques vers les plateformes techniques.

Le MDA est un grand chantier qui s'ouvre. Plus de trente ans après la première crise du logiciel qui a eu lieu dans les années 68, c'est une nouvelle crise qui secoue aujourd'hui le monde de la production de logiciel. Encore une fois la complexité grandissante des systèmes informatiques, leur nécessaire évolutivité et leur couplage de plus en plus fort sur les organisations dans lesquelles ils opèrent ont provoqué une réaction importante et rapide.

Le principe de base du MDA est l'élaboration de modèles indépendants de plateformes (PIM) et la transformation de ceux-ci en modèles dépendants de plateformes (PSM). Les techniques employées sont donc principalement des techniques de modélisation et des techniques de transformation de modèles. L'OMG a d'ailleurs d'ores et déjà commencé à définir des standards dans ce domaine. Les plus connus sont bien sûr UML et XMI mais les autres tels que MOF, CWM ou les profils EJB et CORBA ne sont pas moins indispensables.

D'autres sont à venir, par exemple sur les transformations de modèles ou sur les services web. L'idée est donc séduisante, mais sa mise en œuvre n'en est encore qu'à ses débuts. Il faudra en effet surmonter des défis techniques comme la modélisation des transformations de modèles, mais aussi des défis méthodologiques et stratégiques pour que l'approche MDA puisse s'implanter dans les entreprises.

L'objectif de notre projet est de réaliser une transformation de type modèle vers code :

Nous nous intéressons dans ce mémoire à la transformation des diagrammes de classes UML à des documents XML en suivant l'approche de développement MDA. Par le biais de l'outil ATOM³ nous voulons réaliser un outil qui nous permet par la suite de créer des diagrammes de classes UML.

ATOM³ offre la possibilité de manipuler facilement les grammaires de graphes. Prenant cet avantage nous souhaitons définir une grammaire de graphe pour transformer nos diagrammes à des documents XML. On poursuit l'approche des architectures dirigées par les modèles (MDA) puisqu'en vérité après chaque transformation d'un diagramme de classes UML, créé par notre outil, on obtiendra un document XML qui représente en vérité un modèle textuel indépendant de plateformes (PIM).

Structure de document

Ce document est composé de 3 chapitres :

Nous avons parlé dans le premier chapitre des différents principes de l'approche MDA et de son architecture et en indiquant les différents types de modèles manipulés par MDA, à savoir CIM, PIM et PSM. Ensuite, on a vu aussi les différentes technologies de l'OMG permettant de mettre en œuvre MDA, à savoir MOF, UML, OCL et XMI. En fin on parlera sur les avantages, les inconvénients et le futur de MDA.

Le deuxième chapitre sera consacré à la transformation des modèles qui est considéré comme la clé de la démarche MDA. On a vu aussi les différents formats d'un document XML. Ensuite, nous avons présenté les différentes approches et les règles de transformation. Enfin, nous avons également brièvement présenté ATOM3, l'outil de transformation utilisé dans notre travail.

Le troisième chapitre détaille notre contribution dans laquelle on a commencé par une proposition d'un métamodèle afin d'arriver à la génération d'un outil qui utilise une grammaire de graphes pour la transformation des diagrammes de classe vers des documents XML.

Une conclusion générale et des perspectives de ce travail de recherche termineront ce mémoire.

Chapitre I

CONCLUSION GÉNÉRALE

MDA utilise fortement les modèles que nous avons étudiés au cours de ce travail, à savoir CIM, PIM et PSM. Nous avons présenté brièvement les différentes technologies standards de l'OMG qui permettent de mettre en œuvre MDA, à savoir MOF, UML, OCL et XMI. Nous avons dégagé alors les principaux avantages de MDA qui sont, entre autres, pérennité la productivité et la prise en compte des plates-formes.

Nous avons vue l'état d'art de l'approche MDA, en particulier, les langages de modélisations et ceux de métamodélisation. Nous avons étudié aussi les opérations possibles sur les modèles.

Nous avons développé un méta-modèle représente les concepts et les relations statiques de diagramme de classes d'UML, à savoir la classe, l'association, l'héritage.

Nous avons défini un schéma XML qui validés le document générés après la transformation de diagramme de classes.

Dans ce mémoire nous avons intéressé de présenter une approche pour la transformation des diagrammes de classes UML vers des documents XML. Les documents XML générés en utilisant notre outil de transformation représentent des PIM (Platform Independent Models) puisqu'ils représentent une étape importante dans l'approche MDA. Ils sont validés avec un schéma XML défini par le langage standard XML.

Perspectives

Dans un travail future, nous avons l'intention de développer un outil qui peut utiliser les PIM XML générés afin de générer le code correspondant dans quelques plateformes spécifiques. Nous comptons continuer la transformation des autres diagrammes UML (diagramme état/transition, diagramme de cas d'utilisation, diagramme d'activité, etc ...) en utilisant toujours la technique de transformation de graphes et l'outil ATOM3.

BIBLIOGRAPHIE

- [ATOM3] <http://atom3.cs.mcgill.ca/> date de consultation 04 Mai 2011.
- [BEZ], BEZIVIN J. et BLANC X, *MDA vers un nouveau paradigme(1)*, Développeur Référence, 15 juillet 2002, v2.16, pp. 7-11. Disponible sur : <http://www.lifl.fr/jfiadsma2002/talks/jfiadsma2002-Bezivin.pdf>
- [BEZ2], BEZIVIN J, *Les nouveaux défis des systèmes complexes et la réponse MDA de l'OMG*, [en ligne], 2002, Disponible sur : <http://www.lifl.fr/jfiadsma2002/talks/jfiadsma2002-Bezivin.pdf>
- [BEZ3], BEZIVIN J., *Ecole d'Eté d'Informatique CEA EDF INRIA 2003*, Ingénierie des modèles logiciels, 2003.
- [BLA] BLANC X., *Ingénierie des modèles*, [en ligne], juillet 2004, Disponible sur : <http://www-src.lip6.fr/homepages/Xavier.Blanc/courses/XB-Model-Engineering.ppt>
- [CAR], CARON P-A., *Spécialisation d'un environnement de conception de systèmes flexibles aux Environnements Informatiques pour l'Apprentissage Humain*, Mémoire de DEA, Université des Sciences et Technologies de LILLE, 2 juillet 2003, 53 p.
- [CHE], CHEVAILLIER P., *Techniques avancées de programmation objet*, [en ligne], novembre 2003; Disponible sur : <http://www.enib.fr/info/gl2/cours/pattern.pdf>
- [Dev1] <http://gilles-chagnon.developpez.com/cours/xml/dtd-et-schemas/?page=schemas#LII> date de consultation 06 Avril 2011.
- [Dev2] <http://gilles-chagnon.developpez.com/cours/xml/dtd-et-schemas/?page=dtd#LI-A> date de consultation 06 Avril 2011.
- [FRANCK], FRANCK F, *Langage et méthode pour une ingénierie des modèles fiable*, Thèse de doctorat, Université de Rennes 1, 9 octobre 2006.
- [HADJ], HADJ KACEM M, *Modélisation des applications distribuées à architecture dynamique : Conception et Validation*, Doctorat en informatique, Université Toulouse III -

Paul Sabatier et la Faculté des Sciences Economiques et de Gestion – Sfax, Jeudi 13 Novembre 2008.

- [JUL] JULIOT E., *MDA Model Driven Architecture*, Mémoire de DESS Génie Informatique, Université de Nantes, [en ligne], 10 juin 2002, 16 p., Disponible sur : <http://docs.happycoders.org/orgadoc/dev/mda/toutmda.pdf>
- [Kholladi], Kholladi M, *Une Approche de transformation de la notation BPMN vers BPEL basée sur la transformation de graphe*, Mémoire de Magister en informatique, Université Mentouri Constantine, 2009.
- [Mounir], Mounir G, *Principes et états de l'art de l'approche MDA et applications pour des plates-formes PHP orienté 3-tiers*, Diplôme des Etudes Supérieures Approfondies (DESA) en informatique. Université Mohammed Premier (Maroc), 15 Novembre 2007.
- [OMG] OMG, *MDA Guide Versions 1.0.1*, juin 2003.
- [OMG1] OMG, *Swisslog Software AG*, MDA Success Story [en ligne], 2003, Disponible sur : http://www.omg.org/mda/mda_files/Aonix_Swisslog_SuccessStory.pdf
- [OUM] OUM SACK P-M., *Interopérabilité des Outils d'assistance à l'évolution des systèmes logiciels*, Thèse de doctorat , Laboratoire d'Informatique du Littoral, 26 juin 2003.
- [Raida], Raida E, *Modélisation et Vérification des processus métiers dans les entreprises virtuelles : Une approche basée sur la transformation de graphes*, Thèse de doctorat en informatique, Université Mentouri Constantine, 2008.
- [SRI], SRIPLAKICH P., *Techniques des transformations de modèles basées sur la métamodélisation*, Thèse de doctorat en informatique, Université Pierre et Marie Curie, [en ligne], septembre 2003, 64 p., Disponible sur : <http://modfact.lip6.fr/ModFactWeb/qvt/SimpleTRL.pdf>
- [SUNa] SUNYE G., *Introduction aux patrons de conceptions*, 2009.

- [Sylvain] Sylvain A, *MDA (Model Driven Architecture) principes et états de l'art*,
Diplôme d'ingénieur C.N.A.M. en informatique, CENTRE D'enseignement de Lyon, 05
novembre 2004, Disponible sur :

<http://sylvain.andre.free.fr/MDA/MDAv1.8.pdf>

-[W3] <http://www.w3.org/XML/> date de consultation 04 Avril 2011.

-[Xavier] Xavier B, *MDA en action ingénierie logicielle guidée par les modèles*, 2005.

ملخص

عملنا يهدف إلى تحويل مخططات الفئة UML إلى وثائق XML وفقا لنهج التنمية MDA. ويستند نهجنا على تحويل الرسم البياني واستعمال ATOM3.

ATOM3 يوفر القدرة على التعامل بسهولة مع قواعد النحو الرسم البياني. مع هذه الميزة، اقترحنا نموذج للرسم البيانية لمخططات الفئة UML وقاعدة نحوية تحول تلقائيا مخطط الفئة UML الى وثيقة XML الموافقة له. وما زال نهج الأبنية الموجهة بواسطة النماذج (MDA) لأنه في الحقيقة بعد كل تحول لرسم تخطيطي لفئة UML، التي أنشأتها أداة لدينا، نتحصل على وثيقة XML التي هي في الواقع نموذج نصي مستقل عن منصات (PIM). للتحقق من صحة وثائق XML المتولدة، نضع لنموذجنا الفوقي وثيقة XML تعريف مخطط (XSD).

الكلمات المصنفة: مخططات الفئة UML، تحويل الرسم البياني، MDA، XML، XSD، النموذج الفوقي ATOM3، Python.

Résumé

Notre travail consiste à la transformation des diagrammes de classes UML à des documents XML en suivant l'approche de développement MDA. Notre approche est basée sur la transformation de graphe et l'utilisation d'ATOM3.

ATOM³ offre la possibilité de manipuler facilement les grammaires de graphes. Prenant cet avantage, nous avons proposé un méta-modèle pour les diagrammes de classes UML et d'une grammaire de graphe qui effectue automatiquement la transformation d'un diagramme de classes UML à son document XML correspondant. On poursuit l'approche des architectures dirigées par les modèles (MDA) puisqu'en vérité après chaque transformation d'un diagramme de classes UML, créé par notre outil, on obtiendra un document XML qui représente en vérité un modèle textuel indépendant de plateformes (PIM). Pour valider les documents XML générés, nous ont définis pour notre méta-modèle un document XML Schéma Définition (XSD) correspondant.

Mots clés: Diagramme de classes UML, Transformation de Graphe, MDA, XML, XSD, méta-modèle, ATOM3, Python.

Abstract

Our job is to transform UML class diagrams to XML documents according to the MDA development approach. Our approach is based on graph transformation and use of ATOM3.

ATOM3 offers the ability to easily manipulate the graph grammars. Taking this advantage, we have proposed a meta-model for UML class diagrams and a graph grammar that automatically transform a UML class diagram to its corresponding XML document. It continues the approach of model driven architectures (MDA) since in truth after each transformation of a UML class diagram, created by our tool, we obtain an XML document that represents indeed a textual model of independent platforms (PIM). To validate the generated XML documents, we have set for our meta-model an XML Schema Definition (XSD) matching.

Keywords: UML Class Diagram, Graph Transformation, MDA, XML, XSD, meta-model, ATOM3, Python.