Ministère de l'enseignement supérieur
et de la recherche scientifique
Université de M'sila
Faculté des Mathématiques et de L'informatique
Département d'informatique

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

Domaine : Mathématiques et Informatique
Filière : Informatique
Spécialité : Systèmes d'Informations Avancés

Thème

Transformation automatique du diagramme de classes d'UML vers les réseaux de Pétri (cas d'un processus de production)

Réalisé par :
Oussama Benadel

Dirigé par :
Amel Meliouh

2010 /2011

SOMMAIRE

SOMMAIRE

Chapitre 3

Contribution & Etude de cas

SOMMAIRE

Introduction Générale

Contexte

Capitaliser et réutiliser les artefacts du développement logiciel est crucial pour maîtriser l'augmentation de la complexité des systèmes développés, les nombreux changements des exigences tout au long du développement, ou encore la mise au point de lignes de produits logiciels. Dans l'ingénierie dirigée par les modèles (IDM ou MDE Model Driven Engineering), les transformations de modèles exploitent automatiquement les modèles. Elles sont utilisées pour automatiser certaines étapes du développement : l'introduction de décisions concernant la conception, le ré usinage de modèle, la rétro-conception, ou encore la génération d'une forme exécutable d'un modèle. Ces transformations sont déployées à différentes étapes de développement logiciel. Des industriels les utilisent pour générer du code embarqué(Airbus) ou pour migrer un système en transformant son modèle plutôt qu'en recodant son implantation.

L'architecture dirigée par les modèles ou MDA est une démarche de réalisation de logiciel, proposée et soutenue par l'OMG. C'est une variante particulière de l'ingénierie dirigée par les modèles Le principe de base du MDA est l'élaboration de modèles indépendants de plateformes (Platform Independent Model, PIM) et la transformation de ceux-ci en modèles dépendants de plates-formes (Platform Specific Model, PSM) pour l'implémentation concrète du système.

Aujourd'hui, les systèmes développés sont devenus de plus en plus complexes. Le paradigme de programmation orienté-objet a approuvé de nombreuses avancées et s'implante comme le standard de l'industrie pour le développement logiciel. L'autre standard de l'industrie, UML (Unified ModellingLanguage), est un langage visuel (graphique) spécialement développé pour modéliser proprement un système orienté-objet (OO). Les diverses vues offertes par UML permettent de visualiser plusieurs aspects d'un même système. Ceci permet de mieux gérer la complexité du système.

UML propose un ensemble de notations graphiques standardisées regroupées en treize types de diagrammes complémentaires. Un diagramme donne à l'utilisateur un moyen de visualiser et de manipuler des éléments de modélisation. UML définit des diagrammes structurels et comportementaux pour représenter respectivement des vues statiques et dynamiques d'un système [Muller, 2001]. Le diagramme de classes est le seul diagramme, de la norme UML, à offrir une vision complète de l'ensemble des structures du système à modéliser.

Malgré son succès en étant un langage de modélisation, UML souffre du manque de capacité de vérification et d'analyse [Zhao, 2004]. En effet, les notations semi-formelles et visuelles d'UML peuvent provoquer des inconsistances au niveau des modèles développés.

Beaucoup de modèles ne peuvent réellement être vérifiés en détail, en particulier s'ils sont utilisés pour décrire des systèmes complexes et distribués. Par contre, les réseaux de Petri sont un modèle à événement discret [Ramadge, 1987].Leur théorie permet d'analyser les caractéristiques des systèmes telle que la synchronisation, la concurrence, les conflits, le partage de ressources, les blocages.etc.

Dans ce cas, nous signalons clairement que les transformations entre UML et les réseaux de Petri sont significatives pour l'analyse et la vérification du modèle UML.

Une fonctionnalité essentielle du MDA est la notion de transformation. Une transformation regroupe un ensemble de règles et de techniques pour transformer un modèle en un autre. La transformation des modèles est définie comme étant le processus de conversion d'un modèle d'un système vers un autre modèle [Czarnecki, 2006].

Les travaux réalisés dans ce contexte sont divers, Un code en langage Maude est généré à partir d'UML dans [Flake, 2002], l'approche consiste à modéliser un système par UML et vérifier ses propriétés par le modèle de checking par le biais du code Maude. Les réseaux de Petri représentent le formalisme le plus utilisé pour la vérification, dans [Uzam, 2006] Un RDP est converti à une méthodologie basée sur la logique de passage des jetons (TPLL) afin d'assurer la supervision des systèmes. Les automates d'états finis sont utilisés pour concevoir un superviseur de contrôle pour un système de production dans [Quedraogo, 2006], et sont convertis vers les réseaux de Petri temporisés.

Objectif et démarche

Pour faciliter la modélisation des systèmes qualifiés de complexes et les rendre plus compréhensibles, plusieurs chercheurs ont préconisé un changement de paradigme vers l'orienté objet [Bordbar, 2000]. Dans ce cadre, UML est le langage standard de modélisation le plus utilisé. Dans le domaine de vérification des propriétés de fonctionnement ; les réseaux de Petri ont fait leur preuve. Il s'agit dans notre approche de développer un outil de modélisation par le diagramme de classes d'une part, et de réaliser une transformation du diagramme de classes vers les réseaux de Petri d'autre part.

Pour obtenir une transformation plus générale s'approchant entre UML et les réseaux de Petri, nous avons fait des recherches sur la transformation au niveau méta-modèle. Et pour atteindre un processus de transformation automatique et correct, nous utilisons alors des grammaires de transformation de graphes.

L'outil choisit pour cela est ATOM3 ; car il permet de supporter la multi-modélisation et les transformations de graphes en se basant sur une grammaire. Il est alors nécessaire dans ce cas de comprendre comment cet outil génère automatiquement des modèles UML, et les transformer vers leur équivalent réseau de Petri.

Structure du document

Ce mémoire est organisé en trois chapitres. Le premier chapitre, est un état del'art présentant une présentation du langage UML et les réseaux de Petri .Une attention particulière est portée au diagramme de classes qui nous intéresse dans notre approche.

Le deuxième chapitre est préservé a étudier les notions de base sur la modélisation et les transformations de graphes. Ainsi que la présentation de l'outil ATOM3.

Une étude progressive de notre approche de modélisation et de transformation du diagramme de classes vers les RDPs fait l'objet du troisième chapitre, qui consiste à étudier en détail les étapes de cette transformation. Un exemple applicatif illustrant un processus de production est présenté.

La dernière partie, conclut le mémoire et résume ce que nous avons réalisé dans notre contribution ainsi que les travaux futurs, inspirés de l'approche proposée.

Conclusion Générale

Aujourd'hui, les systèmes développés sont devenus de plus en plus complexes. Le paradigme de programmation orienté-objet a approuvé de nombreuses avancées et s'implante comme le standard de l'industrie pour le développement logiciel. UML (Unified Modelling Language), est un langage visuel spécialement développé pour modéliser proprement un système orienté-objet (OO). Les diverses vues offertes par UML permettent de visualiser plusieurs aspects d'un même système. Ceci permet de mieux gérer la complexité du système. Le diagramme de classe est utilisé pour modéliser l'aspect structurel du système en présentant les éléments composant le système et leurs relations.

Cependant, UML étant un langage à caractère plutôt visuel, il souffre d'un manque de sémantique formelle qui permet de vérifier certaines propriétés de fonctionnement. Cette capacité de vérification est offerte par le formalisme réseau de Petri qui est doté d'une base mathématique très solide.

Dans le cadre de notre travail, nous avons proposé une approche combinant le diagramme de classe avec les RDPs en bénéficiant des deux formalismes.

Contribution

Pour procéder à la combinaison du diagramme de classes et les RDPS, nous avons eu besoin de transformer le premier formalisme vers le deuxième. Dans notre cas, il était donc nécessaire de choisir un outil de transformation des modèles à savoir ATOM3 qui est un outil basé sur la notion de méta-modèle que nous avons abordé dans le deuxième chapitre de ce manuscrit. L'utilisation de l'outil ATOM3 a été avantageuse car il nous a permis d'utiliser un méta-modèle de type Entité /Relation et de générer des outils de modélisation soit pour le diagramme de classes, soit pour les réseaux de Pétri.

Nous avons également introduit la façon dont un outil de transformation de graphes pouvait générer des outils de modélisation à partir des méta modèles associée et créer plusieurs règles permettant de transformer un graphe vers un autre via une grammaire de graphes.

L'approche que nous avons proposé a été appliquée sur un exemple de processus de production illustrant une chaine d'emballage de produits.

Par cette approche, nous avons donc montré qu'il était possible de créer un modèle UML (dans ce cas-ci, un diagramme de classes) avec un outil de modélisation et de transformer ce graphe vers son équivalant réseau de Pétri, avec une grammaire de graphes.

Perspectives

Les résultats obtenus pour notre travail sont très concluants, néanmoins des extensions peuvent être envisagées. Comme notre approche génère un RDP depuis un diagramme de classes, il nous parait intéressant dans une perspective, d'étendre cette méthode de façon à intégrer un outil de vérification des propriétés, exemple l'outil INA dans l'outil ATOM3, cet outil aura comme entrée le RDP résultat et comme sortie la vérification de certaines propriétés fonctionnelles (vivacité, absence de blocage…etc).

Nous annonçons une autre perspective d'amélioration de notre méthode afin d'optimiser le réseau de Petri généré par la grammaire de graphes. En effet dès que le nombre de composants d'un système augmente, la taille du réseau de Petri généré s'élargie et le temps d'exécution de la grammaire de graphe augmente proportionnellement. Il s'agit en fait pour assurer une optimisation de temps d'ajouter à la grammaire de transformation des règles de réduction de ce formalisme Tout en suivant les techniques de réduction des réseaux de Petri.

BIBLIOGRAPHIE

[ATOM3, 2007] http://atom3.cs.mcgill.ca/.

[Bordbar, 2000] Bordbar. B, Giacomini.L, Holding. D.J, "UML And Petri for Design and analysis of distributed systems", Department of Electronic Engineering, School of Engineering. Aston University UK, 2000.

[Clark, 2004] Clark. T, Evans.A, Sammut .P, Willans. J, "Applied Metamodelling A Foundation for Language Driven Development Version 0.1", 2004.

[Czarnecki, 2006] Czarnecki.K, Helsen .S,"Feature-based survey of model transformation approaches", IBM Systems Journal, VOL. 45, NO. 3, 2006.

[De Lara, 2002] De Lara.J , Vangheluwe.H, "AToM3: A Tool for Multi-Formalism Modelling and Meta- Modelling". Lecture Notes in Computer Science 2306, pp.174-18, 2002.

[Dev] www.Developpez.com.

[Esther, 2005] Esther .G, Juan de Lara " Meta-Modelling and Graph Transformation for the Definition of Multi-View Visual Languages ".

[Flake, 2002] Flake.S, "Modeling and Verification of Manufacturing Systems:A Domain-Specific Formalization of UML", C-LAB, Paderborn University, Fuerstenallee 11. 33102 Paderborn. Germany, 2002.

[Gérard, 2005] Gérard .S, " Apprendre à programmer avec Python ". livre, 2005.

[Hettab,2009] HETTAB .A, " De M-UML vers les réseaux de Petri « Nested Nets » : Une approche basée transformation de graphes ", Mémoire de Magistère dans le cadre de l'école doctoral pôle Est, 2009.

[Jean-Marie,2008] JEAN-MARIE .M, " Oracles et qualification du test de transformations de modèles ", Thèse de DOCTEUR DE L'UNIVERSITE DE RENNES 1.

[Kameche,2010] KAMECHE .A, GUERGOUR .M, " Initialisation des RdP à partir Des diagrammes UML ", Mémoire d'ingénieur d'état en informatique option : système informatiques, 2010.

[Laurent,2005] Laurent .P, cours UML, " Ce cours a été écrit en grande partie à partir du site http://uml.free.fr ainsi que du cours de Frédéric Di Gallo ", (CNAM angoulême), 2005.

[Meylan, 2006] Meylan .S, "Etude et comparaison d'outils de transformation de modèles", LIFC, 2006.

[Muller, 2001] Muller .P.A, " De la modélisation objet des logiciels à la meta-modélisation des langages informatiques ", Thèse de HDR présentée devant L'Université de Rennes, 2001.

[Murata,1989] Murata. T, "Petri Nets: properties, analysis and applications", Proc. IEEE, vol.77, pp. 541-580, 1989.

[OMG, 1999] OMG Manufacturing Domain Task Force : STEP and OMG Product Data Management Specifications : A Guide for Decision Makers, Octobre 1999. http://www.omg.org/homepages/mfg/mfgppepdm.htm.

[Quedraogo,2006] Quedraogo .L, Nourelfath.M, Khoumsi.A, "A new method for centralized and modular supervisory control of real-time discrete event systems", Proceedings of the 8th International Workshop on Discrete Event Systems Ann Arbor, Michigan, USA, July 10-12, 2006.

[Raida,2008] Raida .E, "Modélisation et Vérification des processus métiers dans les entreprises virtuelles: Une approche basée sur la transformation de graphes", Thèse de Doctorat en Sciences en Informatique,2008.

[Ramadge, 1987] Ramadge, Wonham.W.M, "Supervisory control of a class of discrete event processes", SIAM Journal on Control &Optimization, pp. 206-230 ,1987.

[Roques, 2007] Roques .P, " UML 2 par la pratique cours et exercices ",5ème édition Edition Eyrolles, 2007.

[Scorletti , 2006] Scorletti.G, Binet.G, "Réseaux de Petri ", Université de Caen Basse Normandie France, http ://www.greyc.ensicaen.fr/EquipeAuto/GerardS/mait Petri.html, 2006.

[Uzam, 2006] Uzam. M, Wonham. M .W ," A Hybrid Approach to Supervisory Control of Discrete Event Systems Coupling RW Supervisors to Petri Nets", International Journal of Advanced Manufacturing Technology, vol. 28, numb. 7-8, pp. 747-760,2006.

[Uzam, 1998] Uzam.M, "Introduction to Petri Nets and Modelling of Discrete Event Systems" chapitre 2. Thèse Ph.D, University of Salford.UK, 1998.

[Vinh Duc, 2005] Vinh Duc.T, " Réseaux de Petri ", Institut de la Francophonie pour l'Informatique Promotion 10 ,2005.

[Zhao, 2004] Zhao. Yu, Yushun.F, Xinxin.B, Wang.Y, Hong.C, Ding.W "Towards Formal Verification of UML Diagrams Based on Graph Transformation",CIM Research Center, Department of Automation, Tsinghua University, Beijing China, 2004.

ملخص

UML: هي لغة موحدةخاصة بالتصميمات، وهي تستخدم لتصميم النظم المختلفة ، وتحديدا في الأنظمة المعقدة. على الرغم من نجاحها، فهي تعاني من عدم القدرة على المراجعة و التحليل. و كثير من النماذج لا تستطيع تحليلها بالتفصيل، على العكس، شبكة بيتري هي النموذج التي لديها أساس نظري متين جدا من أجل تحقيق و تحليل الأنظمة. الموضوع المقترح يسمح لنا بالاستفادة من كلا اللغتين، وذلك بتوفير أداة للقيام بالتحويل الأتوماتيكي لـ diagramme de classes نحو شبكة بيتري.

المرور من لغة UML إلى شبكة بيتري يتم بطريقة أوتوماتيكية موفرة من طرف أداة خاصة بتصميم النماذج المختلفة، تسمى $ATOM^3$.التحويل يتم بتطبيق مجموعة من القواعد البيانية.

هذا العمل سوف يتم تطبيقه على مثال عملي (يمثل عملية تغليف المنتوجات).

الكلمات الرئيسية:شبكة بيتري ، لغة التصميمات UML) (، القواعد البيانية، $ATOM^3$

RÉSUMÉ

UML est un langage visuel de modélisation très riche .Il est utilisé pour la modélisation de divers systèmes et plus précisément les systèmes complexes. Malgré son succès, UML souffre d'un manque de capacité de vérification et d'analyse. Beaucoup de modèles ne peuvent réellement être vérifiés en détail. Par contre, les réseaux de Pétri (RDPs) sont un modèle qui a une base théorique très solide pour la vérification des systèmes.

L'approche proposée permet de bénéficier des deux formalismes, en offrant un outil de transformation automatique du diagramme de classes d'UML vers les RDPs ordinaire.

Latransformation est réalisée par le biais d'une grammaire de transformation de graphes offerte par l'outil ATOM3

Le travail sera appliquésur un exemple d'un processus de productionillustrant une chaine d'emballage de produits

Mots–clés : *UML, ,réseaux de Pétri, grammaire de transformation,ATOM3.*

ABSTRACT

UML is a visual,rich modeling language. It is used for modeling various systems,more preciselycomplex systems. Despite its success, UML suffers from the lack of ability to audit and analysis. Many models can actually be checked in detail. However, Petri nets is a model that has a very solid theoretical basis for verifying systems.

The proposedapproachaims to benefit from both formalisms, providing a tool for automatic transformation of UML class diagram to Petri nets.

The passage from UML's diagrams towards Petri nets is carried out automatically, via a graph grammar by means of a multi-formalisms tool $ATOM^3$.

Thiswork will beapplied toaproduction processexample, that illustrates a chain of packaging products.

Keywords : *UML, Petri net, transformationgrammar, ATOM3.*