

Introduction

contexte

De nos jours, les logiciels sont des outils indispensables dans de nombreux secteurs d'activité. On les utilise dans le domaine de la finance, des transports, la gestion de stocks, l'éducation et bien d'autres. Leur mise en place est toutefois très coûteuse en termes de budget, de ressources humaines et de temps de réalisation et sujette à des risques d'échec de projets. Il devient alors intéressant de produire des logiciels de meilleure qualité à faible coût et rapidement; c'est le but principal de l'ingénierie de ligne de produits logiciels[Software Product Line Engineering Foundations Principles and Techniques]. Elle se base sur le fait que les logiciels développés aujourd'hui ne sont pas nouveaux mais sont des variantes de systèmes déjà développés dans le même secteur d'activité[FORM/BCS].

Au lieu de construire des logiciels en partant de rien (from scratch), on voudrait désormais construire des familles de logiciels en exploitant les connaissances accumulées dans le domaine. Une famille de logiciels ou ligne de produit logiciel (LPL) est un ensemble de logiciels partageant des propriétés communes ainsi que des variantes et satisfaisant des besoins spécifiques pour un domaine donné.

problème et approches connues

L'ingénierie de ligne de produits est constituée de deux étapes principales: l'ingénierie de domaine et l'ingénierie applicative [Software Product Line Engineering Foundations Principles and Techniques, 20]. La première analyse le domaine pour en cerner les propriétés communes aux applications du domaine (commonalities) et les propriétés variantes (variabilities) afin de produire l'ensemble des artefacts logiciels réutilisables. La deuxième quand à elle permet de dériver les logiciels membres de la famille selon les besoins spécifiques en exploitant la variabilité. Ainsi, le développement d'un nouveau logiciel revient à analyser les besoins spécifiques, paramétrer la plateforme support en choisissant les composants nécessaire et générer le code du logiciel. Chacune des étapes de l'ingénierie de ligne de produits se fait en plusieurs phases qu'il convient d'organiser et de structurer. Pour le faire, plusieurs méthodes de développement de lignes de produit existent : FAST(Family oriented Abstraction, Specification and Translation), PuLSE (Product Line Software Engineering) ,ODM (Organization Domain Modeling),DARE (Domain Analysis and Reuse Environment), FODA (Feature Oriented Domain Analysis), FORM(Feature Oriented Reuse Method) etc. La méthode FORM étend la méthode FODA basée sur les caractéristiques métiers (features) qui a été longuement exploitée dans les applications industrielles. FORM a été étendue à son tour pour entre autre élargir son application aux systèmes d'information par la méthode FORM/BCS (Feature Oriented Reuse Method with Business Component Semantics) qui requiert notre attention dans le cadre de ce travail.

question de recherche(sous problème) + résultats attendus

Cependant, la méthode FORM/BCS ne dispose pas encore d'outils support pour permettre son exploitation dans la pratique. Dans le cadre de ce travail, nous voulons contribuer à la mise

en place d'une plateforme support pour la manipulation dynamique des lignes de produit de FORM/BCS. Nous nous attèlerons plus spécifiquement à la conception et à l'implémentation de l'architecture d'un éditeur de composants métier caractéristique à l'aide de l'outil JAVA EMF (Eclipse Modeling Framework).

méthodologie

Pour réaliser ce travail, nous devons tout d'abord comprendre la méthode FORM/BCS en étudiant les propriétés du modèle, puis étudier l'embryon de plateforme support actuelle de FORM/BCS et enfin étendre cet embryon en y ajoutant un éditeur de composants métier caractéristique.

plan

Ce travail est structuré comme suit. D'abord le chapitre 1 présente l'état de l'art des méthodes d'ingénierie de lignes de produits existantes avec leurs points forts et leurs limites. Ensuite le chapitre 2 explique la méthode FORM/BCS en détaillant les caractéristiques du modèle de FORM ainsi que les composants réutilisables de FORM/BCS. Puis, le chapitre 3 présente la mise en oeuvre que nous allons effectuer, il récapitule les étapes de la conception et de l'implémentation de l'éditeur de composant métier caractéristique de FORM/BCS.