



Compilateur LLVM

Langage jouet Kawa

Spécification Technique de Besoin 0.1.5

19 novembre 2014

Auteur(s): Kheireddine BERKANE, Nasser ADJIBI

Version	Date	Changelog
0.1	04/11/2014	Version initiale.
0.1.5	18/11/2014	Définition des cas d'utilisations et des exigences.

Table des matières

1	Objet	2
1.1	Besoins opérationnels	2
1.2	Objectifs techniques	2
1.3	Contraintes et recommandations	2
1.4	Résultats attendus	2
2	Documents applicables et de référence	2
3	Terminologie et sigles utilisés	2
4	Exigences fonctionnelles	3
4.1	Présentation de la mission du produit logiciel	3
4.2	Cas d'utilisation EF_1	4
4.3	Cas d'utilisation EF_2	5
4.4	Cas d'utilisation EF_3	5
4.5	Cas d'utilisation EF_4	5
4.6	Cas d'utilisation EF_5	6
4.7	Cas d'utilisation EF_6	6
4.8	Cas d'utilisation EF_7	6
4.9	Cas d'utilisation EF_8	7
5	Exigences opérationnelles	7
6	Exigences opérationnelles d'interface	7
7	Exigences de qualité	7
8	Exigences de réalisation	8
8.1	Intitulés	8
8.2	Descriptions	8

1 Objet

LLVM est une infrastructure modulaire permettant la réalisation de chaînes de compilation et conçue pour l'optimisation. Elle met en oeuvre une représentation intermédiaire du code qui permet de découpler les langages de l'architecture. Nore objectif est de réaliser, à l'aide de l'infrastructure LLVM, un comilateur pour un langage jouet, que nous appellerons Kawa, ce dernier doit supporter :

- Les classes, les classes abstraites, les interfaces
- L'héritage
- Le polymorphisme
- Le système de types sera composé des types primitifs (int, float, etc.), des classes et des interfaces
- Les instructions de contrôle telles ques (if/else, for, while/do, switch, etc.).
- Les méthodes seront définies de manière identique à Java excepté que les paramètres pourront être préfixés du mot clé VALUE (transmission par valeur au lieu de référence)

1.1 Besoins opérationnels

1.2 Objectifs techniques

1.3 Contraintes et recommandations

1.4 Résultats attendus

2 Documents applicables et de référence

Différents documents de référence :

- Le site LLVM llvm.org.
- Le document de spécification du client Spec1.pdf

3 Terminologie et sigles utilisés

- LLVM : (Low Level Virtual Machine) est une infrastructure de compilateur conçue optimisation à la compilation.
- Kawa : langage jouet qui reprend quelques fonctionnalités de java.
- Clang : compilateur pour le langage c++, son interface de bas niveau se base sur des bibliothèques llvm pour la compilation. Il est utilisé par APPELE.
- GIT : logiciel qui permet de stocker un ensemble de fichiers en conservant la chronologie de toutes les modifications qui ont été effectuées dessus.
- ELF : (Executable and Linkable Format) est un format de fichier binaire utilisé pour l'enregistrement de code compilé (objets, exécutables, bibliothèques de fonctions).
- Ubuntu : Distribution linux sur base Debian.
- C++ : Langage de programmation orienté objet bas niveau.
- POO : Programmation orientée objet.
- Makefile : Fichier regroupant des instructions de compilation avec gestion de dépendances.

4 Exigences fonctionnelles

4.1 Présentation de la mission du produit logiciel

Id	Intitulé	Acteur(s)	Priorité
EF_1	Afficher l'aide	Utilisateur	Indispensable
EF_2	Compiler une application en mode monolithique	Utilisateur	Indispensable
EF_3	Compiler une application en mode partagé	Utilisateur	Secondaire
EF_4	Chosir un nom pour l'exécutable	Utilisateur	Important
EF_5	Compiler des bibliothèques	Utilisateur	Important
EF_6	Afficher la version du compilateur	Utilisateur	Important
EF_7	Définir des dépendances entre sources et classes	Utilisateur	Important
EF_8	Activer l'affichage en couleur	Utilisateur	Important

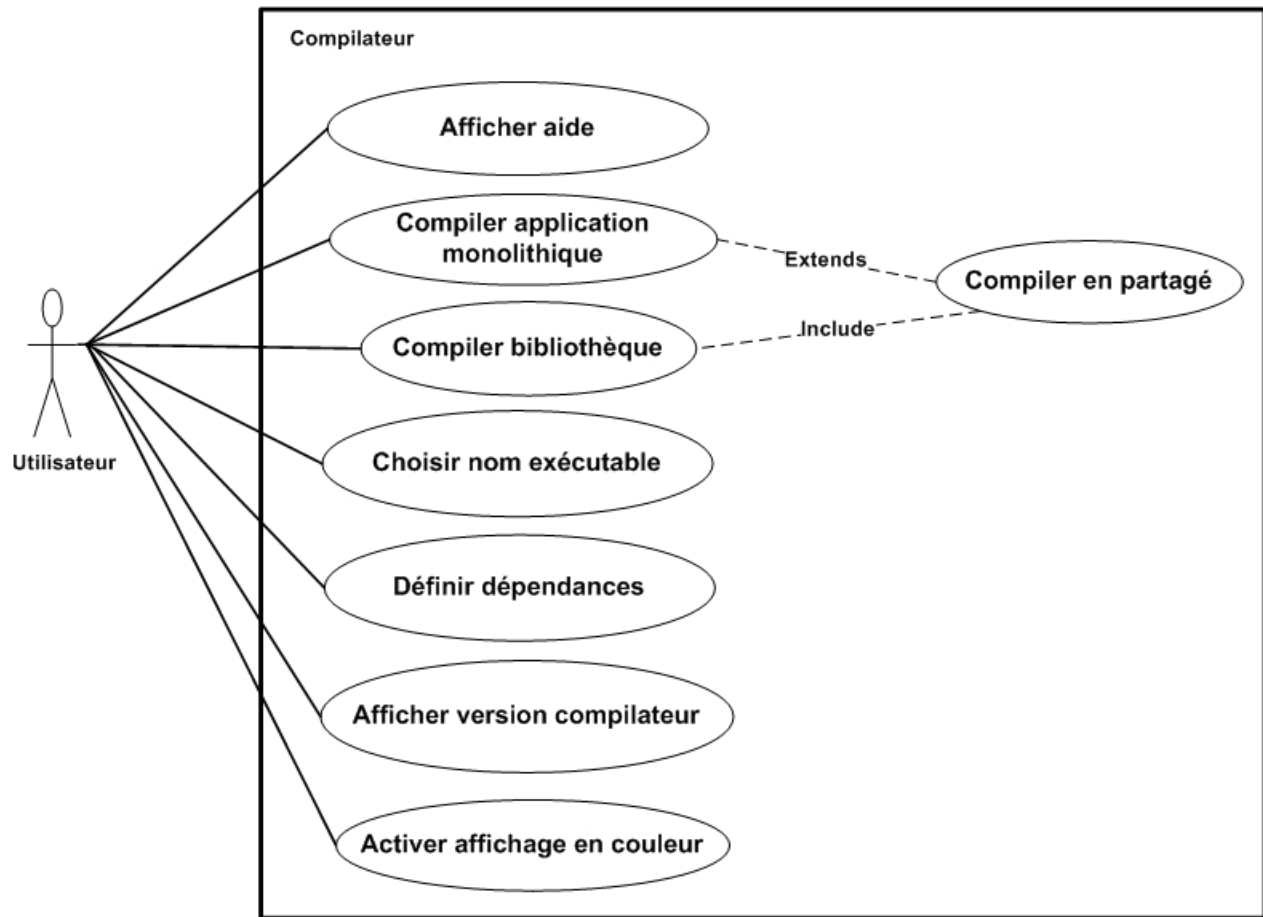


FIGURE 1 – Cas d'utilisations du compilateur kawa.

4.2 Cas d'utilisation EF_1

Nom	Afficher l'aide	
Acteurs concernés	Utilisateur du compilateur	
Description	le compilateur affiche la liste des options du compilateur sur la sortie standard à travers une ligne de commande.	
Préconditions		
Evénements déclenchants	Ligne de commande.	
Conditions d'arrêt	Action utilisateur	
Description du flot d'événements principal:		
Acteur(s)		Système
Flots d'exceptions:		

4.3 Cas d'utilisation EF_2

Nom		Compiler une application en mode monolithique	
Acteurs concernés		Utilisateur du compilateur	
Description		L'utilisateur introduit un ensemble de classes kawa afin de les pré-compiler et de générer le tous dans un seul exécutable.	
Préconditions		Ensemble de fichiers sources respectant la syntaxe du langage kawa.	
Evénements déclenchants		Ligne de commande.	
Conditions d'arrêt		Erreur (syntaxe, fichier introuvable, ..), ou succès de la compilation	
Description du flot d'événements principal:			
Acteur(s)		Système	
Flots d'exceptions:	Abondant provoqué par l'utilisateur.		

4.4 Cas d'utilisation EF_3

Nom		Compiler une application en mode partagé	
Acteurs concernés		Utilisateur du compilateur	
Description		L'utilisateur introduit un ensemble de sources kawa ainsi des exécutables qui auraient été déjà compiler ailleurs,afin de compiler un programme kawa.	
Préconditions		Ensemble de fichiers source d'interfaces respectant la syntaxe du langage kawa.	
Evénements déclenchants		Ligne de commande.	
Conditions d'arrêt		Erreur (syntaxe, fichier introuvable, ..), ou succès de la compilation	
Description du flot d'événements principal:			
Acteur(s)		Système	
Flots d'exceptions:	Abondant provoqué par l'utilisateur.		

4.5 Cas d'utilisation EF_4

Nom	Chosir un nom pour l'exécutable	
Acteurs concernés	Utilisateur du compilateur	
Description	L'utilisateur peut choisir le nom d'exécutable à produire via un ligne de commande.	
Préconditions		
Evénements déclenchants	Ligne de commande.	
Conditions d'arrêt	Action utilisateur.	
Description du flot d'événements principal:		
Acteur(s)		Système
Flots d'exceptions:		

4.6 Cas d'utilisation EF_5

Nom	Compiler des bibliothèques
Acteurs concernés	Utilisateur du compilateur
Description	L'utilisateur introduit l'ensemble des sources de la bibliothèque ce qui implique une compilation en mode partagé. Si le client ne précise pas le nom de l'exécutable, le nom est repris à partir du fichier contenant la méthode main.
Préconditions	
Événements déclenchants	Ligne de commande.
Conditions d'arrêt	Erreur (syntaxe, fichier introuvable, ..), ou succès de la compilation.
Description du flot d'événements principal:	
Acteur(s)	Système
Flots d'exceptions:	Abondant provoqué par l'utilisateur.

4.7 Cas d'utilisation EF_6

Nom	Afficher la version du compilateur
Acteurs concernés	Utilisateur du compilateur
Description	L'utilisateur peut savoir la version du compilateur avec le quel compile ses sources et ses bibliothèques grâce à une ligne de commande.
Préconditions	
Événements déclenchants	Ligne de commande.
Conditions d'arrêt	Action utilisateur.
Description du flot d'événements principal:	
Acteur(s)	Système
Flots d'exceptions:	

4.8 Cas d'utilisation EF_7

Nom	Définir des dépendances entre sources et classes
Acteurs concernés	Utilisateur du compilateur
Description	L'utilisateur peut définir les dépendances utiles pour la compilation entre les sources et des fichiers déjà compilés
Préconditions	
Événements déclenchants	Ligne de commande.
Conditions d'arrêt	Action utilisateur.
Description du flot d'événements principal:	
Acteur(s)	Système
Flots d'exceptions:	

4.9 Cas d'utilisation EF_8

Nom	Activer l'affichage en couleur	
Acteurs concernés	Utilisateur du compilateur	
Description	L'utilisateur peut activer l'option de l'affichage en couleur, afin de décorer les messages renvoyés par le compilateur.	
Préconditions		
Evénements déclenchants	Ligne de commande.	
Conditions d'arrêt	Action utilisateur.	
Description du flot d'événements principal:		
Acteur(s)		Système
Flots d'exceptions:		

5 Exigences opérationnelles

Id	Intitulé	Priorité
----	----------	----------

6 Exigences opérationnelles d'interface

Id	Intitulé	Priorité
----	----------	----------

7 Exigences de qualité

Id	Intitulé	Priorité
EQ_1	les messages d'erreurs renvoyés par le compilateur doivent être explicites, les plus fines possibles et surtout par rapport à ce qu'il s'est passé.	Important

8 Exigences de réalisation

8.1 Intitulés

Id	Intitulé	Acteur(s)	Priorité
ER_1	Nommage des fichiers de sortie	Développeur Kawa	Indispensable
ER_2	Reconnaissance de la grammaire KAWA	Indispensable	
ER_3	Compilation d'application en monolithique	Indispensable	
ER_4	Compilation d'application partagée	Indispensable	
ER_5	Compilation de bibliothèque en monolithique	Indispensable	
ER_6	Compilation de bibliothèque partagée	Indispensable	
ER_7	Compilation de bibliothèque en monolithique	Indispensable	
ER_8			
ER_	Compilation de bibliothèque partagée	Indispensable	
ER_	Fichier de sortie au format ELF	Indispensable	
ER_	Garbage collector	Indispensable	
ER_	Reconnaissance et compilation de classes	Indispensable	
ER_	Reconnaissance et compilation de classes abstraites	Indispensable	
ER_	Reconnaissance et compilation d'interfaces	Indispensable	
ER_	Prise en charge du Polymorphisme	Indispensable	
ER_	Prise en charge de l'héritage de classe	Indispensable	
ER_	Prise en charge de l'implémentation d'interfaces	Indispensable	
ER_	Prise en charge du polymorphisme	Indispensable	
ER_	Prise en charge de l'héritage de classe	Indispensable	
ER_	Prise en charge de l'implémentation d'interface	Indispensable	
ER_	Mécanisme de constructeur	Indispensable	
ER_	Mécanisme de finalisation	Indispensable	
ER_	Reconnaissance et définition d'attribut	Indispensable	
ER_	Indispensable		
ER_	Indispensable		
ER_	Indispensable		
ER_	Indispensable		

8.2 Descriptions

Id	Intitulé
ER_1	La compilation produit un exécutable linux au format ELF
ER_1	Nommage des fichiers de sortie