



# Compilateur LLVM

Langage jouet Kawa

## Spécification Technique de Besoin 0.1

16 novembre 2014

Auteur(s): Kheireddine BERKANE, Pierre-Luc BLOT

Version	Date	Changelog
0.1	04/11/2014	Version initiale.

## Table des matières

<b>1</b>	<b>Objet</b>	<b>2</b>
1.1	Besoins opérationnels . . . . .	2
1.2	Objectifs techniques . . . . .	2
1.3	Contraintes et recommandations . . . . .	2
1.4	Résultats attendus . . . . .	2
<b>2</b>	<b>Documents applicables et de référence</b>	<b>2</b>
<b>3</b>	<b>Terminologie et sigles utilisés</b>	<b>2</b>
<b>4</b>	<b>Exigences fonctionnelles</b>	<b>3</b>
4.1	Présentation de la mission du produit logiciel . . . . .	3
4.2	Cas d'utilisation 1 . . . . .	4
4.3	Cas d'utilisation 2 . . . . .	5
4.4	Cas d'utilisation 3 . . . . .	6
<b>5</b>	<b>Exigences opérationnelles</b>	<b>6</b>
<b>6</b>	<b>Exigences opérationnelles d'interface</b>	<b>6</b>
<b>7</b>	<b>Exigences de qualité</b>	<b>7</b>
<b>8</b>	<b>Exigences de réalisation</b>	<b>7</b>

# 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. Notre objectif est de réaliser, à l'aide de l'infrastructure LLVM, un compilateur 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 que (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](http://llvm.org).
- Le document de spécification du client Spec1.pdf

# 3 Terminologie et sigles utilisés

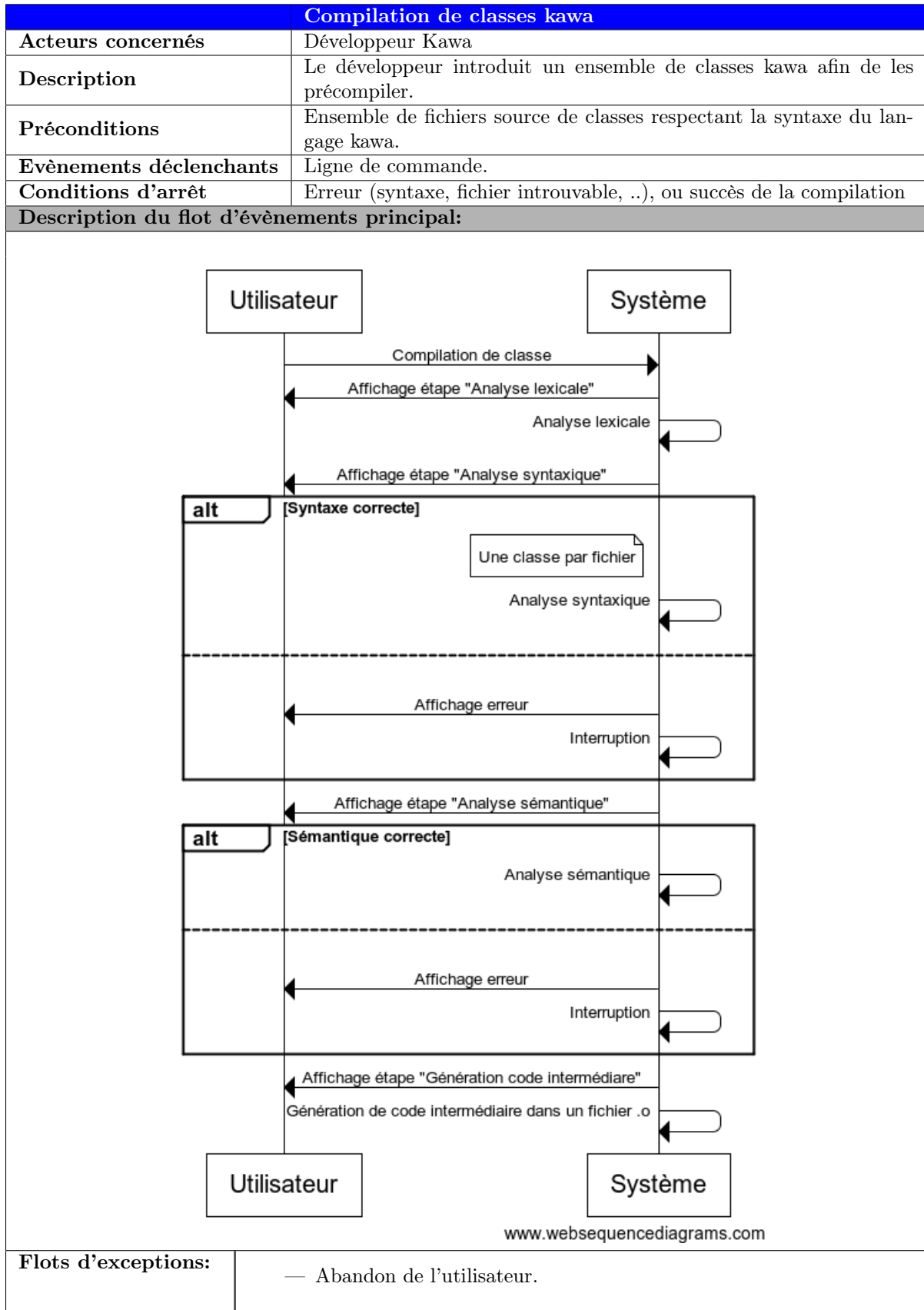
- LLVM : (Low Level Virtual Machine) est une infrastructure de compilateur conçue optimisation à la compilation.
- Kawa : langage jouet qui reprend quelques fonctionnalités de java.
- Clang : compilateur pour le langage c++, son interface de bas niveau se base sur des bibliothèques llvm pour la compilation. 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

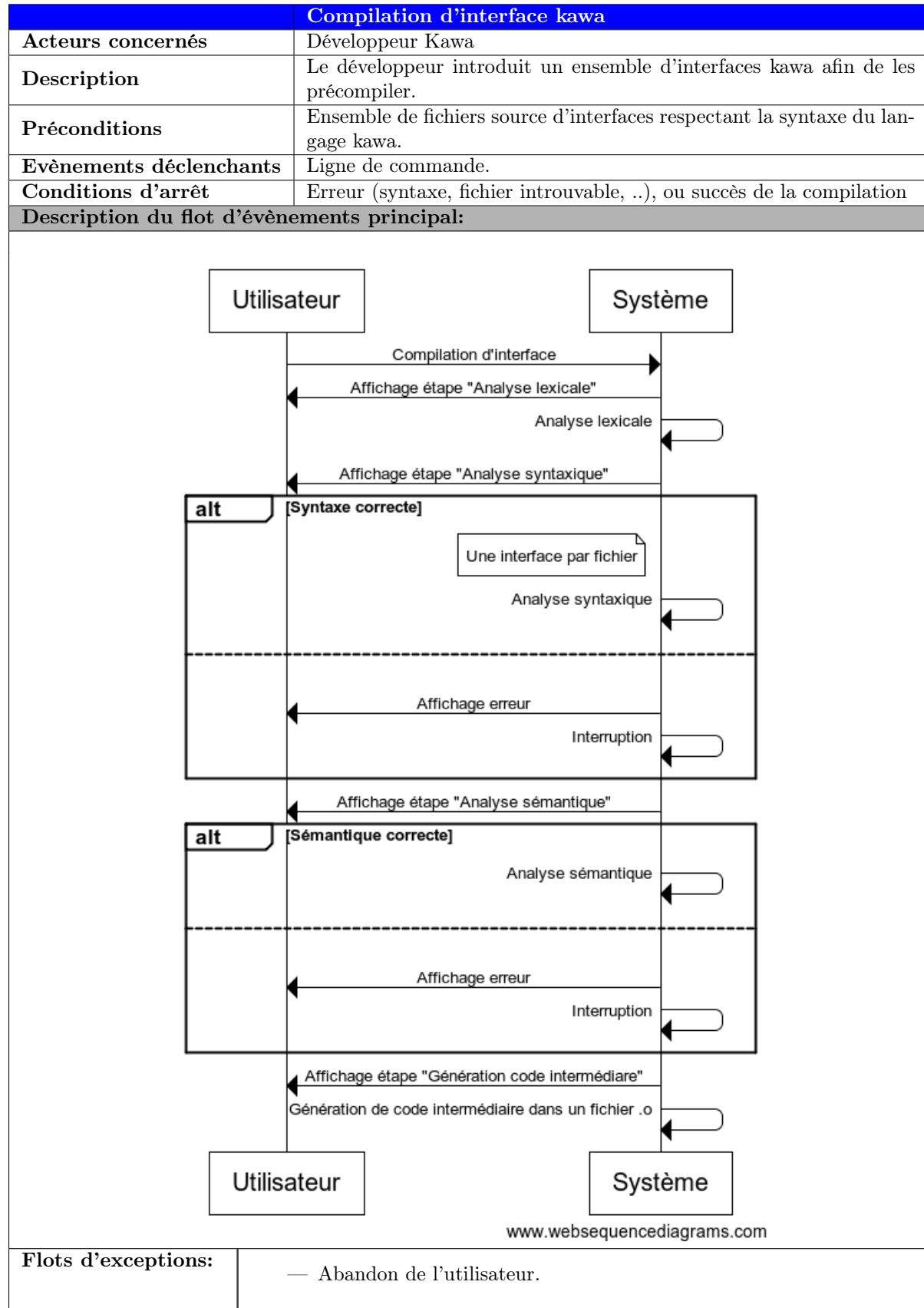
### 4.1 Présentation de la mission du produit logiciel

<b>Id</b>	<b>Intitulé</b>	<b>Acteur(s)</b>	<b>Priorité</b>
EF_1	Compilation code source Kawa	Développeur Kawa	Indispensable
EF_2	Compilation classe	Compilateur	Indispensable
EF_3	Compilation classe abstraite	Compilateur	Important
EF_4	Compilation interface	Compilateur	Important
EF_5	Compiler notion héritage	Compilateur	Important
EF_6	Compiler notion polymorphisme	Compilateur	Important
EF_7	Compiler notion polymorphisme ad hoc	Compilateur	Important
EF_8	Compiler notion polymorphisme de type	Compilateur	Important
EF_9	Compiler bloc de controle	Compilateur	Important
EF_10	Compiler méthodes	Compilateur	Important
EF_11	Compiler méthodes avec paramètres préfixés d'un mot clé VALUE	Compilateur	Important
EF_12	Compiler système de type	Compilateur	Important
EF_13	Compiler affectation	Compilateur	Indispensable
EF_14	Compiler opérateurs arithmétiques sur type primitif	Compilateur	Important

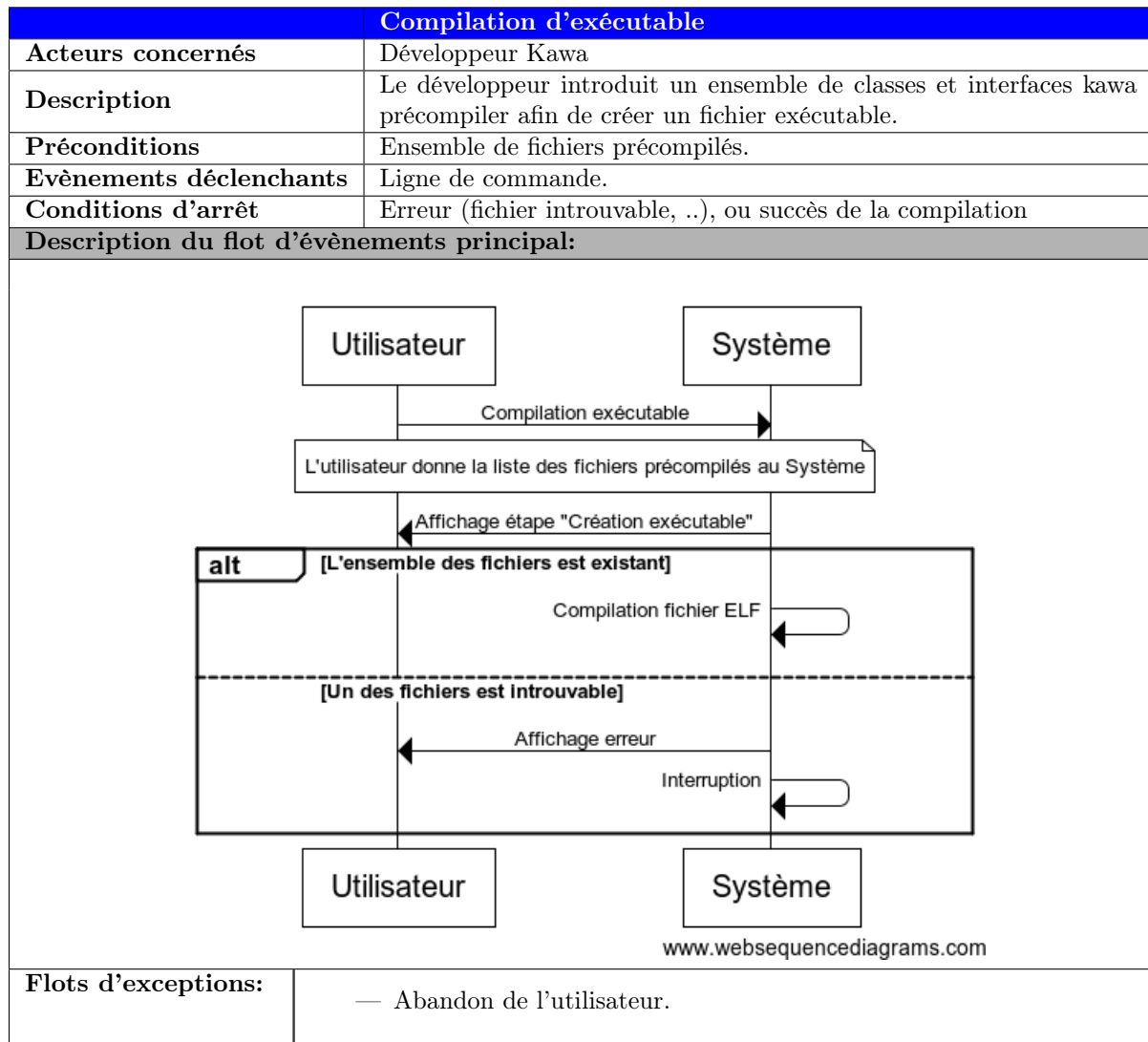
## 4.2 Cas d'utilisation 1



### 4.3 Cas d'utilisation 2



#### 4.4 Cas d'utilisation 3



## 5 Exigences opérationnelles

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

## 6 Exigences opérationnelles d'interface

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

## 7 Exigences de qualité

Id	Intitulé	Priorité
EQ_1	Le compilateur ne plante pas et affiche des messages d'erreurs clairs	Important

## 8 Exigences de réalisation

Id	Intitulé	Priorité
ER_1	La compilation produit un exécutable linux au format ELF	Indispensable
ER_2	Pas d'implémentation des cast	
ER_3	Pas d'implémentation de la générécité	
ER_4	Implémentation en code monolithique	Indispensable
ER_5	Implémentation en code partagé	Optionnel
ER_6	Pas d'attributs dans les interfaces. Seulement des méthodes.	