

Bilan Architecture

Thomas Luquet

November 17, 2010



Documentation control

Group:	Rathaxes
Project name:	Rathaxes
Document Type:	Bilan Architecture
Version:	0.5
Reference:	2012_AA1_FR_Rathaxes
Document status:	Draft

Table 1: Document informations

Version	Date	Name	Summary
0.1	16/11/2010	Thomas Luquet	Premier draft
0.2	17/11/2010	Louis Opter	Fix compilation
0.3	17/11/2010	Louis Opter	Ajout du cartouche
0.4	17/11/2010	David Pineau	Completion de la section Technologie
0.5	17/11/2010	Louis Opter	Reformulations et ajout de notes
0.6	17/11/2010	Luquet Thomas	Reformulation, insertion de graphique

Table 2: Versions history

Contents

1	Rappel du projet	4
1.1	Qu'est ce que Rathaxes ?	4
1.2	Structure du projet	4
2	Diagramme	5
2.1	Description Fonctionel	5
2.2	Diagramme détaillé	6
3	La technologie Rathaxes	7
3.1	Un héritage	7
3.1.1	Un compilateur	7
3.1.2	Une série de modèles	7
3.2	Un langage	8

Chapter 1

Rappel du projet

1.1 Qu'est ce que Rathaxes ?

Rathaxes est un ensemble d'outils permettant de simplifier l'écriture de pilote de périphérique. Le projet permet de générer un code source écrit en C pour Linux, Windows 7 et openBSD à partir d'un seul fichier de description de pilote.

C'est un projet à visée scientifique distribué sous licences libres.

Le projet Rathaxes 2012 est une amélioration de l'EIP réalisé en 2009. Il incorpore de nouvelles fonctionnalités comme l'asynchronicité. Rathaxes 2012 sera capable de générer un pilote de souris USB et celui d'une carte son qui servient à prouver que le concept fonctionne.

1.2 Structure du projet

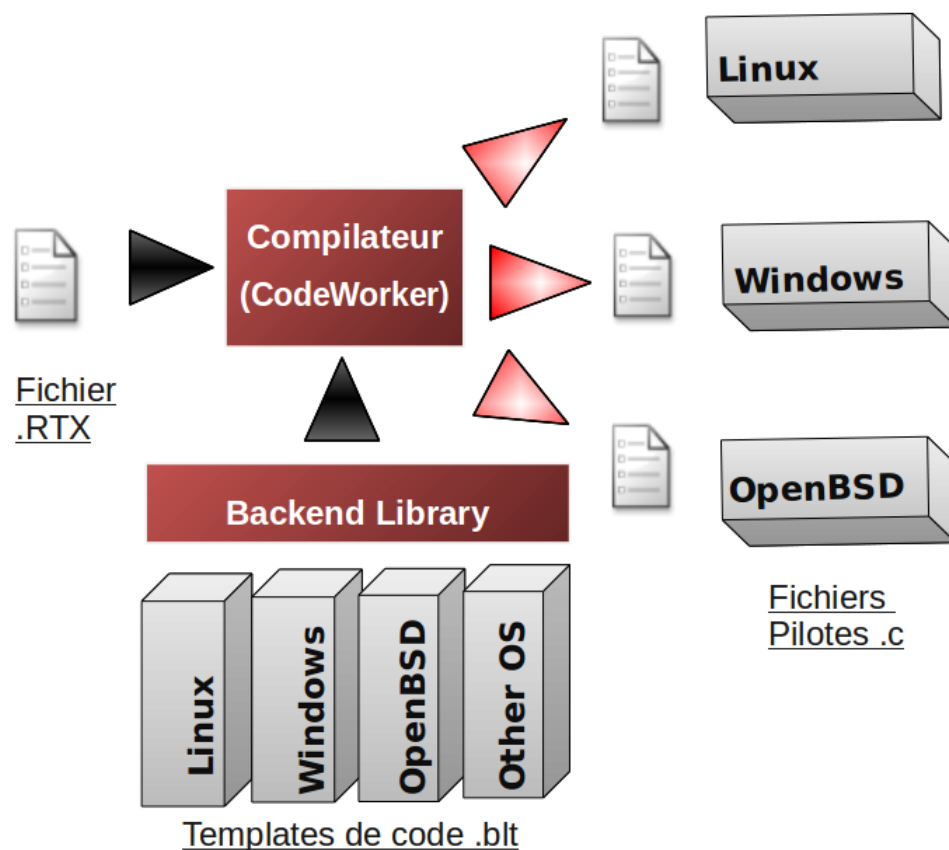
Rathaxes est divisé en trois parties :

1. Le langage Rathaxes : un langage dédié (DSL) utilisé pour décrire un pilote ;
2. La black-librairie : Elle permet l'interfaçage entre le DSL et le compilateur ;
3. Le compilateur : Il transforme les fichiers Rathaxes (.rtx) —à l'aide de la black-librairie— en fichier .c spécifiques au système d'exploitation choisi.

Chapter 2

Diagramme

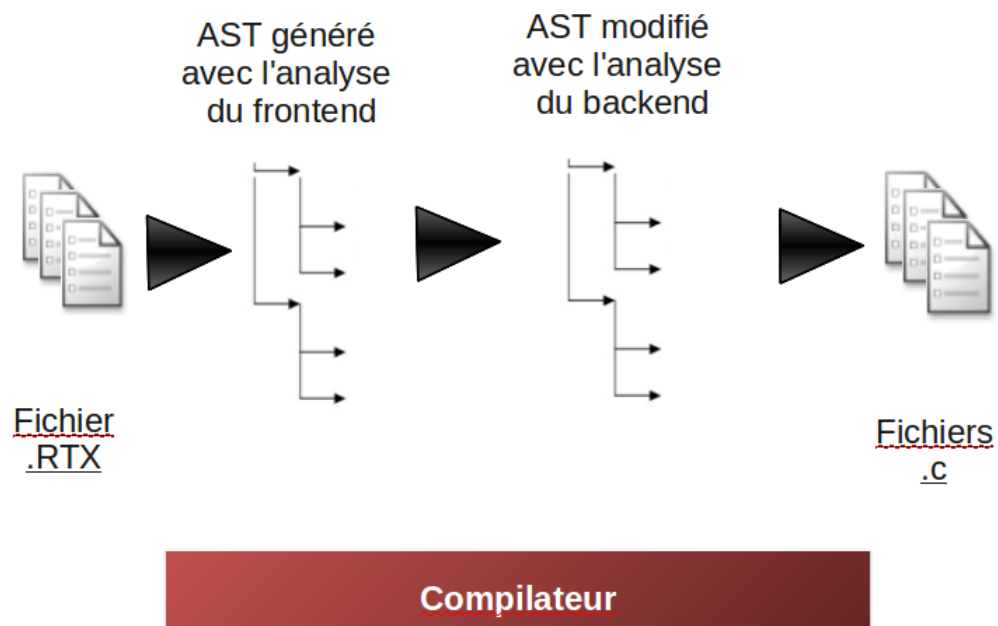
2.1 Description Fonctionel



Rathaxes est un compilateur. Il utilise le langage Codeworker pour l'analyse du DSL ainsi que la génération d'AST. Enfin, Codeworker utilise cette AST, pour générer le code C du driver. Afin de pouvoir générer du code C à partir du DSL de Rathaxes le compilateur utilise une bibliothèque de modele appelé

Backend Library.

2.2 Diagramme détaillé



Actuellement le compilateur Rathaxes issu de la promotion 2009 fonctionne ainsi :

1. Il transforme les fichier .RTX (DSL) en arbre de syntaxe abstraite (AST).
2. Puis il parcourt cette arbre. En fonction des mots clefs rencontré, le compilateur sélectionne des fichiers BLT issus de la backend Library.
3. Enfin il génère des fichiers de pilote de périphérique selon l'arbre reconstruit à partir des modèles de la BLT.

Notre objectif est de modifier le comportement du compilateur dans le but rendre les modifications futur du langage plus simple à implémenter. Pour cela l'étape de transformation de l'AST en fichier .C sera segmenter en 2 étapes :

La 1er sera la modification de l'AST généré à partir du DSL. L'équipe insérera des bouts d'AST généré à partir de fichiers de la Backend Librarie activé selon les mots clefs du DSL.

La 2em partie sera la transcription de l'arbre finale en code C.

Chapter 3

La technologie Rathaxes

3.1 Un héritage

Un grand nombre de choix technologiques ont été effectués par l'équipe 2009. Ils avaient besoin d'une technologie permettant d'implémenter facilement un compilateur, qui puisse utiliser des modèles de code.

3.1.1 Un compilateur

Pour écrire le compilateur, l'outil retenu par la première équipe est CodeWorker. CodeWorker est un langage interprété dont la syntaxe s'inspire la notation EBNF. L'interpréteur CodeWorker est distribué gratuitement sous la licence libre LGPL.

CodeWorker, présente l'avantage d'être facilement abordable et permet une analyse syntaxique et lexicale avancée, tout en permettant la génération de code en se basant sur l'arbre de syntaxe abstraite généré par le code précédemment analysé

De plus, la syntaxe proposée pour le langage de script de CodeWorker est syntaxique relativement proche de la syntaxe Backus-Naur Form sous sa version étendue. Sa syntaxe proposée pour la manipulation d'arbre syntaxique et de génération de driver est dérivée du C. Langage que l'équipe Rathaxes 2012 maîtrise.

3.1.2 Une série de modèles

L'objectif du projet est de générer des drivers en C pour différents systèmes d'exploitation. Ceci doit être fait en indépendance des connaissances spécifiques à un OS. Il est donc nécessaire de posséder un système de modèles utilisables pour générer du code C pour chaque système spécifique.

C'est aussi pour cela que l'équipe 2009 a choisi CodeWorker. C'est l'outil le plus pratique. En effet, c'est la solution la plus simple pour avoir des modèles qui permettent de générer du code C. CodeWorker permet de réduire la quantité de travail nécessaire du côté du backend de Rathaxes et ce, notamment pour la partie parsing.

3.2 Un langage

Forts de l'expérience acquise par l'équipe 2009 de Rathaxes, l'objectif de notre groupe est de rechercher et implémenter de nouveaux concepts dans Rathaxes. Parmi ceux-ci, s'illustre notamment les notions d'asynchronicité, de DMA (Direct Memory Access), d'IRQ (Interrupt ReQuest) et enfin l'ensemble des concepts liés à l'utilisation de BUS tels que le PCI dans les pilotes de périphériques.

Un autre objectif de notre équipe est d'apporter des améliorations dans le fonctionnement même de Rathaxes. Jusqu'alors, le code C était généré par activation des modèles en fonction du contenu de l'arbre de syntaxe abstraite. Nous désirons changer cela, pour d'une part rendre le coeur du compilateur plus générique, et permettre de tester efficacement chaque étape d'implémentation du langage.

Pour cela, nous allons conserver le CodeWorker, qui par sa simplicité, nous permettra de modifier le coeur du compilateur, afin par la suite de pouvoir facilement intégrer de nouveaux concepts et mots clefs au compilateur. Par ailleurs nous pourrions intégrer plus aisément des contributeurs externes au projet.