



ETNA  
Projet de Fin d'Étude 2005-2007  
RimElse  
Cahier des charges

© Copyleft 2006, ELSE Team

23 mars 2006

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Présentation du projet</b>	<b>3</b>
2.1	Une distribution “Évoluable” . . . . .	3
2.2	L’usage . . . . .	4
2.2.1	<i>etna_e</i> étudiant à l’ETNA . . . . .	4
2.2.2	<i>adm_p</i> professeur d’informatique . . . . .	5
<b>3</b>	<b>Les Buts de RimElse</b>	<b>7</b>
3.1	La conception de RimElse . . . . .	7
3.1.1	Les réalisations . . . . .	7
3.2	Les choix techniques . . . . .	8
<b>4</b>	<b>Conclusion et choix moraux</b>	<b>10</b>

# Chapitre 1

## Introduction

Aujourd’hui, les utilisateurs de solutions informatiques sont de plus en plus nomades. Mais, ils sont de ce fait confrontés à un nouveau problème : Comment disposer d’un système d’exploitation qui corresponde à leurs besoins sans ordinateur portable ?

Actuellement, le monde du logiciel libre propose des solutions mobiles tels que les “live cd” ou les systèmes sur clés usb. Or, ceux-ci ont encore des inconvénients tels que la rapidité d’accès aux données système présentes sur le media source (ex : le cdrom du live cd). Ainsi que la persistance des données créés lors de l’utilisation et pour finir l’impossibilité pour l’utilisateur de faire évoluer son système.

Dans le cadre de leur projet de fin d’étude à l’ETNA (PFE), le groupe ELSE, ETNA Linux System Engineers, composé de six étudiants a décidé de fournir une solution au problème précédent. Cette solution, la distribution RimElse fait donc l’objet du présent cahier des charges.

Ce cahier des charges comprend donc, premièrement, une description des différentes caractéristiques de la distribution tels que le Run In Memory et la gestion de l’“évoluabilité”. Deuxièmement, il décrit les différentes réalisations qui seront effectuées par le groupe ELSE afin de bâtir la distribution RimElse. Et dans une dernière partie, il expose les choix techniques fait par l’équipe de développement.

## Chapitre 2

# Présentation du projet

### 2.1 Une distribution “Évoluable”

Dans le soucis de résoudre les problèmes posés plus haut nous avons donc créé “RimElse GNU/Linux”.

“RimElse GNU/Linux” comme son nom l’indique est donc une “Distribution GNU/Linux Run In Memory”. Voyons tout d’abord ce qu’est une distribution GNU/Linux.

La définition de wikipedia.fr<sup>1</sup> est très parlante :

Les distributions rassemblent les composants d’un système GNU / Linux dans un ensemble cohérent et stable facilitant son installation, utilisation et maintenance. Elles comprennent donc le plus souvent un logiciel d’installation et des outils de configuration. Il existe de nombreuses distributions, chacune ayant ses particularités propres, certaines sont dédiées à un usage spécifique (pare-feu, routeur, grappe de calcul, ...) d’autres à un matériel spécifique. Les grandes distributions sont généralistes et/ou pour le bureau.

Le Run In Memory, quand à lui, est basé sur un principe simple, l’intégralité du système est chargé en mémoire puis exécuté depuis cette dernière.

Il y a biensûr dans le monde du libre des solutions existantes, mais nous nous proposons d’apporter un plus non négligable, créer un système “Évoluable”, concaténation d’Évolutif et de Malléable.

---

<sup>1</sup><http://fr.wikipedia.org/wiki/Distribution.Linux#D.C3.A9finition>

Évolutif car nous voulons apporter la possibilité à l'utilisateur de changer les versions de ses logiciels ainsi que d'en ajouter ou d'en retirer comme sous une distribution classique.

Malléable car nous voulons que ce système puisse être adaptable aux besoins de chacun tout en satisfaisant le plus grand nombre. Sans oublier le fait que le système est "Run In Memory" et donc forcément optimisé.

"RimElse GNU/Linux" doit être le système qui tient dans votre poche !

## 2.2 L'usage

Il est fort aisé de présenter des grands concepts sans en déterminer "l'usage".

RimElse va être utilisable dans plusieurs contextes, nous allons en décrire deux.

Le premier est le cas d'un étudiant de l'ETNA.

### 2.2.1 *etna\_e* étudiant à l'ETNA

Être étudiant à l'ETNA entraîne un certain nombre de contraintes sur son système :

- Avoir un lien pour se connecter à l'intranet,
- Disposer d'un lecteur Flash inclus dans son navigateur,
- Disposer d'un client SSH pour se connecter aux plateformes de l'école,
- Avoir un compilateur et un debugger C pour travailler sur ses projets,
- ...

Un étudiant qui se retrouve connecté sur un ordinateur autre que le sien ne disposera pas forcément de l'intégralité de ses outils. De même s'il ne possède pas d'ordinateur portable.

C'est là que RimElse prend tout son sens, avec une simple clé USB (et un CDROM dans le cas où le PC hôte ne peut pas démarrer sur USB) l'étudiant *etna\_e* disposera de son environnement d'étude avec lui, dans sa poche. Si l'espace disponible sur la clé, il pourra alors y sauvegarder ses données.



FIG. 2.1 – Schéma d'utilisation pour un étudiant

### 2.2.2 *adm\_p* professeur d'informatique

*adm\_p* doit préparer un TP C Unix à ses étudiants, mais il ne dispose pas de tels systèmes d'exploitation dans son établissement.

Avec une RimElse qu'il adapte à ses besoins (client samba, compilateur C, ...) il charge un système Unix sur toutes les machines de la salle avec une seule et unique clef usb qu'il aura préparé.



FIG. 2.2 – Schéma d'utilisation pour un professeur

## Chapitre 3

# Les Buts de RimElse

L'équipe ELSE a fixé un certain nombre d'objectifs à atteindre pour RimElse. Le premier de ces objectifs est d'avoir une séquence de démarrage, rapide et une reconnaissance matérielle la plus exhaustive possible.

Ensuite RimElse devra pouvoir fonctionner de manière autonome en RAM, sans accéder à son support initial.

Une synchronisation du système devra être possible, à n'importe quel moment afin de garder les modifications apportées.

RimElse devra comme toute distribution disposer de ses propres outils de configuration et de gestion des paquets binaires. Pour ces derniers nous avons pour but d'affiner la gestion des dépendances en fournissant la possibilité de faire des patches de binaires. Ainsi pour ajouter un support à un paquet, par exemple un support LDAP à Postfix, il suffira d'appliquer un patch et non pas de supprimer le paquet et d'installer celui qui contient le support. Pour cela nous devons redéfinir un type de paquet binaire que nous nommerons de manière temporaire “.reb” (RimElse Binary). Ces binaires devront être optimisés comme pour un système embarqué, afin d'alléger le plus possible la distribution.

Enfin, le dernier point qui n'est pas des moindres, nous fournirons un outil de génération de systèmes personnalisés. Chacun pourra ainsi générer un système selon ses propres besoins. L'étudiant *etna\_e* n'a pas les mêmes besoins que *adm\_p* mais tous deux ont besoin de RimElse.

### 3.1 La conception de RimElse

#### 3.1.1 Les réalisations

Afin d'atteindre les objectifs précédents, plusieurs jalons ont été établis.

### **La séquence de démarrage**

Tout d'abord il est nécessaire d'optimiser la séquence de démarrage, il en découlera la création d'un script de reconnaissance matériel le plus léger et le plus performant possible. C'est à dire reconnaissant un nombre important de périphériques. Toujours dans une problématique de rapidité, il sera impératif de redéfinir l'utilitaire de lancement de services.

### **La gestion des binaires**

RimElse est une distribution Run In Memory, cela implique un traitement spécial des binaires. De ce fait, il convient d'implémenter un nouveau format de paquets ainsi qu'un gestionnaire adapté.

### **Le gestionnaire de configuration**

Le gestionnaire de configuration permettra à l'utilisateur de modifier les paramètres de sa distribution en fonction de ses besoins, par exemple la configuration de l'interface graphique ou du réseau. Il s'interfacera donc avec le gestionnaire de paquets.

### **Le générateur de distribution**

Pour pousser encore plus loin la personnalisation de sa distribution, il sera possible pour l'utilisateur de générer une RimElse selon ses critères. RimElse proposera plusieurs types de profils prédéfinis, par exemple un profil étudiant ETNA ou encore développeur Java. De plus, l'utilisateur pourra définir de nouveaux profils.

Enfin, pour être en adéquation avec les différents points cités et pour répondre aux problématiques posées, la RimElse sera disponible via différents supports tels que les Compact Flash, CD-Rom ou au sein de partages réseaux.

## **3.2 Les choix techniques**

Pour répondre aux problèmes de lisibilité, disponibilité et surtout de pérennité, les choix suivants se sont imposés.

- Concernant la documentation, deux outils ont retenus notre attention :
- $\text{\LaTeX}$  pour toute la partie communication sur le projet,
  - “Doxygen” qui va permettre de générer de la documentation à partir des commentaires dans le code.

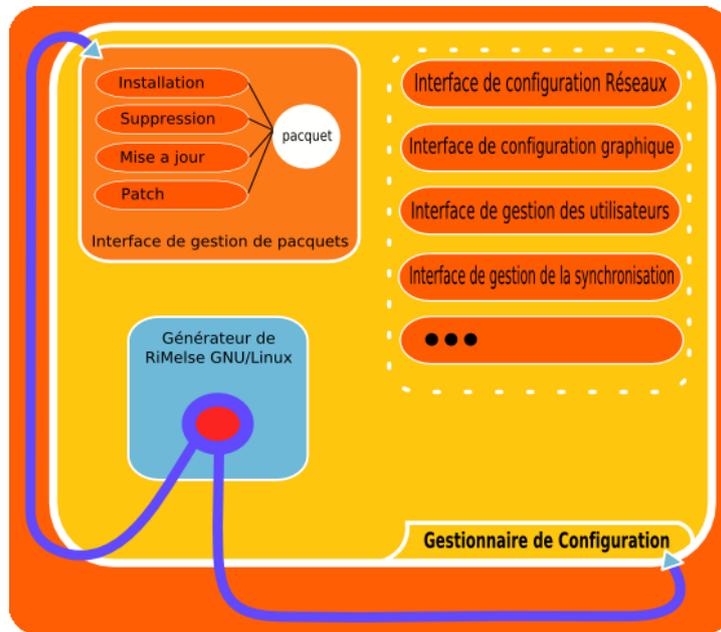


FIG. 3.1 – Schéma montrant l'agencement générale de la configuration

Ces outils permettent le rendu de documents sous différents format à la manière d'une documentation d'API standard.

D'autre part toute modification du code source, ou de la documentation, doit pouvoir être suivie par l'ensemble des membres du groupe, aussi il a été décidé de déployer le gestionnaire de version "git". Celui-ci est, entre autre, utilisé pour la gestion du code source du noyau Linux.

Au-delà de l'aspect de contrôle, ce processus de travail permet une implication globale de chaque membre quelque soit sa tâche au sein du projet. En effet il est possible de revenir en arrière si une erreur a été commise dans le code.

En ce qui concerne les différents développements, nous avons principalement opté pour deux langages :

- Ruby, langage orienté objet, performant et rapide,
- C, incontournable dans le monde Unix,
- Nous aurons probablement à mettre en place des scripts shell.

Les noyaux les plus récents sont les plus fournis en support matériel, par conséquent nous n'utiliserons que les noyaux 2.6 ou supérieurs.

## Chapitre 4

# Conclusion et choix moraux

Simplicité, disponibilité, efficacité complétés par une pluralité des supports de distribution sont les mots d'ordre de RimElse.

Dans cette optique, le groupe ELSE développera des solutions de gestion de distribution adaptées à ses besoins. On notera la création de gestionnaires d'initialisation, de paquets, de configuration et d'un générateur de distribution.

Pour finir, la license GPLv2 garantira la liberté de son projet.

# Bibliographie

- [1] Définition de distribution : [http://fr.wikipedia.org/wiki/Distribution\\_Linux](http://fr.wikipedia.org/wiki/Distribution_Linux)
- [2] L<sup>A</sup>T<sub>E</sub>X : <http://www.latex-project.org/>
- [3] Doxygen : <http://www.stack.nl/~dimitri/doxygen/>
- [4] Git : <http://www.kernel.org/git/>
- [5] Ruby : <http://www.ruby-lang.org/en/>

# Table des figures

2.1	Schéma d'utilisation pour un étudiant . . . . .	5
2.2	Schéma d'utilisation pour un professeur . . . . .	6
3.1	Schéma montrant l'agencement générale de la configuration .	9