

ISAIP

18, rue du 8 Mai 1945

49000 St Barthélémy d'Anjou



Dossier Final

Développement d'une application de personnalisation de système
d'exploitation : "OS A La Carte"

7 Février 2007

Mohammed BADISS, Jérôme CHARLOT

2^{ème} année CPI

Promotion 2005 – 2007

Sous la responsabilité de :

M Alexandre LEPRIEULT ;

M François MANIÈRE.



SOMMAIRE

1	Introduction	2
2	Convention de rédaction	2
3	Présentation du projet	2
4	Présentation du procédé de fabrication	3
5	Technologies utilisées	3
6	Méthode de développement	4
7	Répartition des tâches	5
8	Planning effectif	5
9	Bilan du projet	6
10	Difficultés rencontrées et solutions apportées	7
11	Apports de notre projet	7
12	Pourquoi ce projet a été réalisable	8
13	Apports personnels	9
14	Évolutions possibles	9
15	Conclusion	11
16	Remerciements	11
A	Annexes	12
A.1	Validation du Sujet	13
A.2	Cahier des Charges	16
A.3	Dossier de Spécifications	29
A.4	Dossier de Conception	57
A.5	Dossier Protocoles de Tests	87
A.6	Journal de Bord	103
A.7	Dossier Économique : Simulation de la mise en production de l'application	110
	Annexe A – Custom Install CD	120
	Annexe B – Demande d'hébergement du projet sur SourceForge.net	133
	Annexe C – Licence d'utilisation GPL	139
	Annexe D – Article de presse 01net	145
B	Acronymes	147
C	Glossaire	148
D	Bibliographie	153

1 Introduction

Dans le cadre de notre deuxième année de notre formation Chef de Projet International en Informatique et Réseaux à l’**ISAIP**, nous avons réalisé un Projet de Fin d’Études. Ce dossier relate l’expérience que nous avons eu avec ce projet.

Le projet a pour but la **personnalisation** d’un système d’exploitation à partir d’un site Internet, plus particulièrement de la distribution Linux **Ubuntu**. Il s’agit d’un projet **réel** ayant pour nom “**OS A La Carte**”.

Dans ce dossier nous détaillerons ce qui entoure le projet mais surtout la gestion de projet avec la méthode de développement Agile orientée tests que nous avons choisi : cette méthode s’appelle l’**Extreme Programming**.

2 Convention de rédaction

Pour réaliser l’ensemble des dossiers nous avons mis au point une convention d’écriture.

Les mots importants sont écrits en **gras**.

Les mots qui sont définis dans le glossaire sont soulignés.

Les paragraphes qui font référence à des adresses internet sont suivis de la référence numérotée [1].

3 Présentation du projet

Notre projet a pour rôle de rendre un système d’exploitation **pleinement** utilisable. Tout d’abord, il convient de détailler toutes les étapes nécessaires pour arriver à cette **finalité**.

Lorsqu’une personne souhaite utiliser son ordinateur, elle commence par **installer** son système d’exploitation. Pour cela, elle insère le média d’installation, il s’agit généralement d’un CD ou d’un DVD, elle démarre sur le média (“boot” sur le média) après le BIOS. Elle installe alors le système d’exploitation, une fois cette étape achevée, elle installe ensuite les **périphériques** avec des pilotes. Elle continue par l’installation des **applications** selon ses besoins et enfin elle termine par la **configuration** de celles-ci.

C’est dans ce contexte que notre projet prend naissance. En effet, notre projet a pour rôle de prendre en compte toutes ces actions et de construire un système d’exploitation dit “**clé en main**” ou **personnalisé**, pour qu’il corresponde le plus possible aux exigences et besoins de l’utilisateur.

Notre système d’exploitation se base sur une distribution Linux nommée **Ubuntu**.

Notre projet s’appelle “OS A La Carte”, à la carte, faisant référence aux restaurants, ou le client **choisit** les éléments de son menu.

Notre **objectif** est de mettre à disposition pour l’utilisateur une image du DVD d’installation au format ISO qu’il devra installer par la suite.

4 Présentation du procédé de fabrication

Le procédé de fabrication se déroule en **trois** phases, qui sont les suivantes :

1. La première phase est la **sélection** de la configuration depuis notre interface Web. Cette opération est la seule effectuée par l'utilisateur. Le temps de réponse l'interface web dépend principalement de la bande passante du client, et de notre serveur. Mais avec les connexions actuelles, la technologie ADSL étant largement répandue, ce temps est **négligeable** (à peine quelques secondes).
2. Une fois la phase de personnalisation terminée. La phase de **génération** du DVD d'installation, réalisée par notre application, peut exiger environ **3** minutes sur le serveur dédié (un Pentium 4 ou équivalent, dans des conditions idéales de charge du processeur).
3. Finalement, la phase de **téléchargement** de l'image du DVD est la plus **exigeante** au niveau temps, car comme la première phase, elle dépend de la bande passante que dispose le client mais surtout celle que nous allouons pour le serveur de téléchargement.

5 Technologies utilisées

Pour réaliser ce projet, nous avons utilisé la distribution Linux **Ubuntu** dans sa version **6.10**, sortie au mois d'octobre 2006. Pour tout ce qui concerne l'interface de sélection, nous avons utilisé donc un serveur web **Apache**, une base de données **MySQL** et enfin le langage de script **PHP**.

Pour ce qui concerne la partie de traitement, plus particulièrement la **génération** de l'image ISO du DVD nous avons utilisé le langage interprété **Python**. Et enfin pour l'élaboration des fichiers de références nécessaire à la création des images ISO de “base”, nous avons employé le langage de traitement par lot **Bash**.

Pour développer en **PHP** et en **Python**, nous avons utilisé l'**Environnement de Développement Intégré (EDI)** Eclipse et notamment le plugin **pydev**.

Cette documentation a été réalisée grâce au langage de composition de document **LaTeX** pour générer un fichier final en **PDF**. Les schémas ont été réalisés avec **OpenOffice.Org**, **Microsoft Visio** et **Objecteering**.

6 Méthode de développement

Notre méthode de développement est une méthode **Agile**. Il s’agit d’une méthode qui fait intervenir le client ce qui permet une grande **réactivité** par rapport à un cycle plus traditionnel (modèle en “V” notamment).

La méthode **Agile** que nous avons choisi s’appelle l’**Extreme Programming**, elle se caractérise par l’usage **intensif de tests**. On parle de tests de **non-régression** qui sont effectués sur le logiciel.

Les tests de **non-régression** ont pour but de vérifier que la **nouvelle** version du logiciel fonctionne de la même manière que la version **précédente**, et que les évolutions de la nouvelle version n’ont pas amené directement ou indirectement un mauvais fonctionnement.

Ainsi, dès que nous ajoutons de nouvelles modifications à notre programme, par exemple le support de nouveaux paquets ou l’ajout de fonctionnalités pour les paquets, nous **redéroulions** à chaque fois tous les tests de **A à Z**.

Ces tests sont menés par des **scénarii** de tests, qui sont définis avant le codage de l’application. Ainsi nous avons passé une bonne partie de notre temps à coder le programme de tests qui doit prendre en compte un fichier en entrée pour apprendre un scénario.

Comme le **client** intervient lors du développement, la documentation des dossiers de spécification et de conception s’en retrouve plus **légère**, on définit seulement ce qui est essentiel, par exemple les interfaces ne seront pas détaillées au maximum mais on préférera les enchaînements principaux. Comme il s’agit d’un cas d’étude, nous les étudiants, devons jouer également le rôle du client.

De plus, cette méthode **Agile** est également caractérisée par une procédure dite **itérative**, en effet, on produit un cycle **complet** de développement : spécification, conception, test puis validation, cet enchaînement est réalisé pour chaque ajout de fonction, puis lorsqu’on ajoute de nouvelles fonctionnalités, on réalise à nouveau toutes ces étapes.

Il est important de préciser que cette méthode préconise un développement “**simple**” (en anglais le proverbe “Keep It Simple, Stupide” : **KISS**), d’aller directement à l’**essentiel** et de tester.

Pour conclure, cette méthode permet de **s’adapter aux changements** et donc de faire face aux demandes que le client pourrait demander à la dernière minute.

Ce cycle a été choisi, car il s’agissait d’un projet ambitieux, il était important de réaliser un premier essai sous forme de **prototype** pour savoir si le projet était réalisable et si il l’était dans les temps que l’école nous accordait.

7 Répartition des tâches

Au tout début du projet, M Jérôme CHARLOT s’est occupé du **prototypage** qui a permis de vérifier que le projet était **réalisable** dans le temps accordé.

Une fois que nous étions fixé quant à la “faisabilité” du projet et que le projet a été validé auprès des professeurs responsables, M Jérôme CHARLOT s’est occupé à travailler et coder sur les tests de **non-régression**, qui est l’élément **central** de toute la démarche de notre projet.

Pendant ce temps là, M Mohammed BADISS s’est attaché faire la demande d’hébergement auprès de la plate-forme de développement **SourceForge**, de plus il a développé le site de présentation du projet qui est consultable à l’adresse Internet suivante : <http://osalcproject.sourceforge.net/index.html>.

Ensuite, les deux étudiants, à savoir M Mohammed BADISS et M Jérôme CHARLOT se sont occupés tous les deux **ensemble** de l’élaboration du cahier des charges, de la rédaction du dossier de spécification et enfin de l’écriture des dossiers de conception et des protocoles de tests.

Enfin au niveau du **codage** de l’application et de la réalisation des **tests unitaires**, nous nous sommes réparti les tâches.

M Mohammed BADISS a développer le module qui permet de consulter les informations sur les paquets des dépôts (le module “Watch and Notify”), le module qui permet d’aller chercher les informations sur les paquets qui sont sur les dépôts (le module “Feed”) et enfin a développé l’interface de sélection de l’utilisateur (le module “Select”).

M Jérôme CHARLOT, quant à lui, a conçu le module qui permet de générer l’image au format ISO du DVD d’installation (le module “Remaster”), du module qui paramètre le DVD d’installation (le module “Preseed”) et également des autres petits modules.

Notre méthode de développement **Agile** : l’**Extreme Programming**, qui préconise le développement en binôme, a pour conséquence que si un développeur est bloqué par un bogue, l’autre peut lui venir en aide.

8 Planning effectif

On remarquera sur le diagramme du planning effectif, que les tests de “faisabilité”, par la réalisation d’un **prototype**, ont été effectués avant la validation du projet par l’équipe pédagogique.

On notera également que la première chose qui a été réalisée est le développement des tests de **non régression**.

Puis ensuite intervient le développement des modules en eux-mêmes.

De plus, dès le début, il a été prévu la possibilité de donner suite au projet avec la demande d’hébergement sur la plate-forme de développement **SourceForge** (CF Annexe Dossier sourceforge).

9 Bilan du projet

Le projet est **fonctionnel**, comme nous avons choisi un cycle de développement **itératif**, le stade de la version **1.0**, où toutes les fonctionnalités de “base” sont implémentées, est atteint. Il s’agit d’un projet “vivant”, il sera continué et de nouvelles fonctionnalités seront ajoutées et seront alors testées grâce aux tests de **non-regression**.

Au jour d’aujourd’hui, il faut **3** minutes pour générer l’image au format **ISO** du DVD, si l’utilisateur décide d’installer tous les profils d’applications qui lui sont mis à sa disposition. Le DVD occupera alors un espace de **1,1 Go**.

En ce qui concerne le temps de téléchargement, on peut dresser un tableau du temps d’attente nécessaire à l’utilisateur pour télécharger l’image du DVD d’installation :

Type de connexion	Débit effectif descendant	Temps de téléchargement
1 Mega	130 Ko/s	5h12min
8 Mega	1 Mo/s	18min
20 Mega	2,5 Mo/s	8min

FIG. 1 – Tableau – Estimation du temps de téléchargement du DVD d’installation selon les différents types de connexion.

Ainsi au total, le temps d’attente pour l’utilisateur ayant une très bonne connexion sera de **13min** alors que pour un utilisateur ayant une connexion dite “moyenne” sera de **5h18min**. Cette action peut paraître lente pour une connexion “lente”, mais il est à noter que cette opération n’est effectuée qu’une seule fois.

Les améliorations du logiciel porteront sur l’amélioration de l’interface de sélection et la mise en place d’une aide pour aider l’utilisateur dans ses choix.

10 Difficultés rencontrées et solutions apportées

Les difficultés rencontrées sont de deux ordres, **technique** d’abord puis **organisationnelle**.

- Difficulté technique.

Nous avons deux programmes de conception hétérogène à faire communiquer, un script de sélection réalisé en **PHP** (le module “Select”) et un programme de personnalisation de l’image du DVD d’installation développé en **Python** (le module “Remaster”).

Or l’objectif était d’automatiser le processus complet de personnalisation, faire en sorte que l’utilisateur n’ait aucune action à faire lorsque sa sélection est terminée.

Pour solutionner cette difficulté, nous avons utilisé la fonction **popen()**[35] du langage **PHP** qui permet de lancer notre script **Python** et lui avons passé en paramètre l’**identifiant** de la sélection de l’utilisateur dans le but de transmettre sa sélection.

- Difficulté organisationnelle.

Un membre de l’équipe n’avait pas Internet, il a été nécessaire de prendre cette **limitation** en considération.

En effet, le développement de notre application requiert une aide disponible sur Internet pour réaliser des opérations plus techniques.

Nous avons surmonté cette étape en permettant au développeur de travailler en étant **hors ligne**, à savoir en **téléchargeant** les documentations nécessaires pour le travail qui était à faire. Nous avons ainsi téléchargé les manuels de prise en main des langages de développement **PHP** et **Python**.

11 Apports de notre projet

- Pour l’entreprise tout d’abord.

Selon un article de **01net** (CF Annexe “Article de presse - 01net - “Peut-on encore se passer d’Eclipse ?””), les entreprises n’ont pas de distributeur pour l’environnement de développement (**EDI**) qu’elles utilisent.

Ainsi, elles sont donc obligées, soit de maintenir eux même les versions de l’environnement de développement, soit de laisser les développeurs se débrouiller par eux-même.

Notre application s’inscrit dans ce contexte et propose ainsi des modules répondants aux différents **besoins**. Par exemple on trouvera un module **Eclipse général**, un module **Eclipse UML** pour créer de la documentation en norme UML, un module **Eclipse Unit** pour réaliser les tests unitaires.

- Pour le particulier, un système “**clé en main**” permet d’avoir un système complètement **fonctionnel**.

Selon les tâches qu’il souhaitera réaliser, nous lui mettons déjà à disposition des **profils d’applications**.

Nous proposons une solution de “**montage vidéo**”, qui regroupera tous les utilitaires nécessaires à cette tâche comme l’acquisition de la vidéo, programme de montage vidéo et programme de création de DVD.

L’utilisateur peut également avoir une solution “**multimédia**”, qui prend en compte plus de 200 formats vidéos et 50 formats audios.

De plus, si cette personne est un **développeur**, nous lui proposerons d’installer tous les outils nécessaires pour la programmation. Ainsi nous proposons déjà, des profils “Développement Web” (PHP), “Développement Python” et “Développement Java”.

12 Pourquoi ce projet a été réalisable

Ce projet a été réalisable car nous sommes à un **moment clé** au niveau des évolutions technologiques.

D’une part, nous sommes à une période charnière où le monopole de l’éditeur de logiciel Microsoft est de plus en plus **remis en cause** par l’amélioration de systèmes d’exploitations concurrents tels que **Mac OS X** ou encore **Linux**.

D’autre part, ce projet a été réalisable, car nous avons utilisé un langage de développement qui a fait ses preuves, en effet **Python** a été créée en 1989 et il s’agit d’un langage de développement de “haut niveau” qui permet de manipuler directement tous les outils systèmes issus du monde **Unix** et **Linux**.

De plus, on constate une évolution très récente de la **vitesse** des connexions haut débit de type ADSL. En effet depuis deux ans, on voit que la taille de la bande passante a très largement augmentée. Ce qui fait que dans le meilleur des cas, il faut plus que **quelques minutes** pour télécharger notre image du DVD d’installation au format ISO. Le temps d’attente de l’utilisateur se voit ainsi très réduit.

Et enfin, notre projet utilise la virtualisation logicielle dans le but de simuler une machine et de réaliser nos tests. Ce concept est assez nouveau et il faut remonter à deux ou trois ans pour voir son apparition.

13 Apports personnels

Par rapport à notre formation, nous avons utilisé les compétences de **développement**, et notamment avec le développement en **langage procédural**. De plus, nous avons utilisé nos cours de “**Théorie des systèmes d’exploitation**” avec l’utilisation des outils Linux.

Au niveau du pôle “**Réseau**”, nous avons mis en application également nos connaissances en réseau et notamment au niveau du fonctionnement des **services** comme le paramétrage d’un serveur Web ou d’une base de données.

Ce projet nous a permis de nous **confronter** avec le métier de Chef de projet qui est au coeur de notre formation. Cette expérience nous a permis de **gérer** notre temps, en effet, il est important de prendre en compte que nous n’avons pas que le projet à réaliser mais également d’autres tâches à gérer en parallèle, d’où l’intérêt de prévoir et d’établir un planning prévisionnel.

14 Évolutions possibles

Comme il s’agit d’un projet **Open Source**, nous prévoyons de nous joindre au projet existant qu’est **Ubuntu**, une distribution Linux communautaire, sans doute constituer une branche **parallèle** de ce qui existe actuellement.

Pour ce faire, nous avons dès le début pris les devants et avons demandé un hébergement sur la plate-forme de développement de projets libres **SourceForge**[1], nous avons pour cela fait une demande (CF Annexe B – “Demande d’hébergement du projet sur SourceForge.net”) et développé un site qui présente notre projet en langue anglaise. Le site est consultable à l’adresse internet suivante : <http://osalcproject.sourceforge.net/index.html>.

De plus sur cet espace, nous mettons nos codes sources disponibles sous **licence libre** (CF Dossier gpl), pour que d’autres personnes puissent y contribuer et enrichir notre projet. Dans un avenir proche, nous ouvrirons notre projet et nous utiliserons les **outils de développement communautaire**, comme le “dépot” (outil de “versionning” notamment) pour y mettre notre code source, utiliser le programme de rapport de bogues et également les mailing lists pour être en contact avec les utilisateurs.

Toutefois, nous souhaitons garder une certaine **indépendance**, et nous réfléchissons actuellement sur la possibilité de lancer nous même sur Internet le service de personnalisation de système d’exploitation. Pour cela nous avons détaillé les **coûts** nécessaires à la **location de machines** pour héberger et fournir nous même le service (CF Annexe – Dossier Économique : Simulation de la mise en production de l’application). Il ne s’agit pas d’un simple hébergement comme pour les pages Web puisque nous utilisons des “**primitives**” de bas niveau du système d’exploitation (comme la création de dépôt, la signature de dépôt,...).

Enfin, pour savoir si notre projet est réellement intéressant, nous créerons sans doute une **association** loi 1901, et ferons une demande de dons pour couvrir les frais de location des serveurs pour l’hébergement.

Au niveau de l’orientation générale, nous nous sommes toujours orienté vers l’**utilisateur final** et avons donc modifié d’abord les “couches” les **plus visibles** (les plus hautes) pour l’utilisateur, comme la

configuration du navigateur Internet, des raccourcis clavier généraux.

Cependant en lisant la presse informatique récemment, nous nous sommes aperçu que certains programmes comme les **Environnements de Développement Intégré** (ou **EDI**), comme par exemple Eclipse, n'étaient pas encore bien intégrés dans les distributions Linux avec tous leur plugins existants.

Nous avons pour cela créer déjà deux profils d'applications, nommés respectivement **alc-edi-java** et **alc-edi-python** qui justement ont pour rôle d'installer Eclipse pour l'utiliser et développer des applications en **Java** et en **Python**. Ainsi selon les besoins des entreprises nous pourrions étendre notre gamme de profils d'applications et proposer de nouveaux profils d'applications (Eclipse pour les tests unitaires ou encore Eclipse UML). Cette idée provient **d'un article de presse** (CF Annexe D – “Article de presse 01net”).

Dans un tout autre domaine, il existe des entreprises qui vendent des machines personnalisables, comme par exemple **DELL** [2], sur le site, nous pouvons choisir la **configuration matérielle** de l'ordinateur portable comme par exemple l'écran, la puissance du processeur, la quantité de mémoire vive, nous pourrions ainsi leur proposer d'ajouter la personnalisation au niveau logiciel.

Nos recherches ont donc mis en avant un fabricant américain de portables : **System76**[3] qui vend des portables équipés du système d'exploitation Linux et plus particulièrement de la distribution Linux Ubuntu, celle dont nous nous sommes basées pour réaliser notre projet.

Ainsi, nous pourrions proposer à cette entreprise d'**intégrer** notre projet à leur solution, ce qui permettrait pour leur entreprise de personnaliser à la fois le **matériel** et le **logiciel** pour leurs différentes gammes de produits.

15 Conclusion

Ce projet est un projet très **ambitieux** car il touche à un niveau peu commun qu’est le système d’exploitation. Il est de plus très **enrichissant** tant du point de vue **technique** aussi bien en développement qu’en réseau, que du point de vue de la **gestion de projet**.

De plus, ce projet aura été l’occasion de sortir d’un cycle de développement **traditionnel** et d’apprendre une nouvelle méthode de développement. En effet, nous avons choisi une méthode **Agile** et plus particulièrement la méthode appelée **Extreme Programming**. Cette méthode fait participer le client, et est donc plus “réactive”, elle adaptée au travail en binôme et à la particularité de faire un usage intensif des **tests** et de privilégier la simplicité.

Et enfin ce projet sort très largement du cadre scolaire puisqu’il s’agit d’un projet **réel**. De plus il sera continué. Nous avons **plusieurs pistes** quant à son avenir, que ce soit la **sponsorisation** par une entreprise qui vend des ordinateurs et souhaite personnaliser son offre logiciel, l’intégration du projet sur lequel tout notre travail repose : la communauté **Ubuntu** ou encore la création de notre propre **association** pour développer ce projet.

16 Remerciements

Nous tenons a remercier particulièrement M Alexandre LEPRIEULT qui nous a fait confiance pour lancer ce projet, qui est assez ambitieux, qui sort très largement du cadre **scolaire** mais également pour ses **conseils** et l’**encadrement** du projet.

Nous remercions également M Pierre LAPLAZE, pour ses **conseils** pour la réalisation du dossier économique.

Et enfin nous n’oublions pas non plus M MANIÈRE qui nous **assistera** et nous **évaluera** lors de la soutenance finale.

A Annexes

ISAIP

18, rue du 8 Mai 1945

49000 St Barthélémy d'Anjou



Dossier Validation du Sujet

Développement d'une application de personnalisation de système
d'exploitation : "OS A La Carte"

9 Novembre 2006

Mohammed BADISS, Jérôme CHARLOT

2^{ème} année CPI

Promotion 2005 – 2007

Sous la responsabilité de : M^{elle} Amandine
DUFFOUX.

M Alexandre LEPRIEULT ;

M Pierre BODIN ;

M Redouane DJELOUAH.



1 Présentation du sujet

Notre projet a pour but la **personnalisation** d’un système d’exploitation depuis une interface web.

Il aura pour objectif de simplifier la configuration du système d’exploitation en générale pour les utilisateurs dits “débutants”.

Nous nous appuyerons sur le système d’exploitation Linux, et plus particulièrement Ubuntu qui est une distribution Linux basée sur Debian.

2 Définition du projet

Comme il s’agit d’un projet ambitieux, il est important de délimiter les actions que nous réaliserons dans ce projet.

Tout d’abord notre cible sera principalement les couches les plus élevées du système d’exploitation : comme la personnalisation du bureau (raccourcis, agencement des menus, ...) ou encore l’ajout d’applications additionnelles. Ainsi notre cible sera l’utilisateur final et donc le particulier.

Pour simplifier, d’un point de vue technique, on s’attardera plus particulièrement sur l’architecture **i386** et seulement pour les ordinateurs dits “Desktop” (ceux de bureau).

3 Outils utilisés

Nous utiliserons les langages interprétés **Python** et **PHP**, une base de données relationnelle **MySQL** ainsi que tous les outils mis à disposition par la distribution **Debian** pour la création et la modification de paquets, la mise en place d’un dépot et enfin la création d’une image ISO.

4 Notions du cours

Ce projet fera intervenir d’une part la modélisation **UML**, la programmation **procédurale**, la création d’une base de données relationnelle mais également une application des cours de “Théorie des Systèmes d’Exploitation”.

5 État du projet et modèle de développement

Nous avons déjà commencé le projet, avec la recherche de documentation pour la personnalisation du DVD d’installation. Nous avons obtenu de l’aide sur le site officiel **Ubuntu**.

Dans le but de lever les points techniques **critiques** qui pourraient nous bloquer par la suite, nous nous sommes déjà mis à la pratique à savoir, la “customization” du DVD d’installation avec quelques scripts bash déjà fournis, nous avons déjà réalisé un premier prototype avec les scripts déjà fournis.

Nous utiliserons un cycle de développement Agile : l’Extreme Programming. Nous élaborerons un noyau dur tout d’abord puis nous ajouterons par la suite de nouvelles fonctionnalités.

ISAIP
18, rue du 8 Mai 1945
49000 St Barthélémy d'Anjou



Cahier des Charges

Développement d'une application de personnalisation de système
d'exploitation : "OS A La Carte"
30 Novembre 2006

Mohammed BADISS, Jérôme CHARLOT
2^{ème} année CPI
Promotion 2005 – 2007

Sous la responsabilité de :
M Alexandre LEPRIEULT ;



Table des matières

1	Présentation du projet.	3
1.1	Origine.	3
1.2	Nature du travail demandé.	3
1.3	Les différentes personnes impliquées.	3
2	Le besoin.	4
2.1	Identification et formulation du besoin ou but du produit.	4
2.2	Définition des cibles.	4
2.3	Définition des objectifs.	4
3	Contexte.	4
3.1	Étude de l'existant.	4
3.2	Étude des concurrents.	5
3.3	Environnement technique.	6
3.4	Ressources mises à disposition.	6
3.5	Respect d'un règlement.	6
3.6	Suites prévues.	6
4	Directives particulières.	7
4.1	Contraintes de développement.	7
4.2	Contraintes méthodologiques.	7
4.3	Contraintes de sécurité.	7
5	Expression fonctionnelle du besoin.	9
6	Exigences qualité.	9
7	Aspect financier.	10
8	Remise de la documentation.	10
9	Planning prévisionnel.	11
10	Maintenance.	12
11	Assistance.	12

Table des figures

1	Diagramme UML – Cas d’utilisation du projet OS A La Carte.	4
2	Schéma – Différentes étapes du procédé d’installation.	5
3	Tableau – Historique des sorties des différentes versions d’Ubuntu.	8
4	Tableau – Alertes de sécurité d’Ubuntu pour sa version « Edgy Eft ».	8
5	Schéma – Domaine d’intervention du projet OS A La Carte.	9
6	Schéma – Découpage du projet OS A La Carte en modules.	11

Contenu détaillé du cahier des charges fonctionnelles (Norme Afnor X50 151). Le cahier des charges est un élément constitutif qui unit les deux partenaires. Le cahier des charges constitue le contrat technique du travail à faire. C'est le cahier des charges qui oriente la démarche de conception et qui permet de justifier, en aval, les choix techniques.

1 Présentation du projet.

1.1 Origine.

Notre projet aura pour rôle de rendre un système d'exploitation **pleinement** utilisable. Tout d'abord il convient de détailler toutes les étapes nécessaires pour arriver à cette finalité.

Lorsqu'une personne souhaite utiliser son ordinateur, elle commence par installer son système d'exploitation. Pour cela, elle insère le média d'installation, il s'agit généralement d'un CD ou d'un DVD, elle démarre sur le média (« boot » sur le média) après le BIOS. Elle installe alors le système d'exploitation, une fois cette étape achevée, elle installe ensuite les périphériques avec des pilotes. Elle continue par l'installation des applications selon ses besoins et enfin elle termine par la configuration de celles-ci.

C'est dans ce contexte que notre projet prend naissance. En effet, notre projet a pour rôle de prendre en compte toutes ces actions et de construire un système d'exploitation dit « clé en main » ou personnalisé, pour qu'il corresponde le plus possible aux exigences et besoins de l'utilisateur.

Notre projet s'appelle « OS A La Carte », à la carte, faisant référence aux restaurants, où le client choisit les éléments de son menu.

1.2 Nature du travail demandé.

Dans un premier temps, nous prendrons en considération tous les **outils** mis à disposition comme la gestion des paquets, la prise en main du **procédé d'installation** d'Ubuntu et enfin la modification de l'image ISO.

Dans un second temps, nous définirons les selections que l'utilisateur pourra réaliser avec notre application.

Notre travail constitera à **adapter** le système d'exploitation selon les demandes de l'utilisateur final.

1.3 Les différentes personnes impliquées.

Les personnes impliquées sont tout d'abord, le tuteur du projet à savoir M Alexandre *LEPRIEULT*, ensuite les deux personnes en charge du projet, étudiants à l'institut de l'ISAIP, à savoir M Mohammed *BADISS* et M Jérôme *CHARLOT*.

2 Le besoin.

2.1 Identification et formulation du besoin ou but du produit.

Le projet a pour but de personnaliser un système d'exploitation, une distribution Linux depuis un page Internet.

2.2 Définition des cibles.

Ce projet vise comme public les utilisateurs *finaux* et éventuellement les utilisateurs plus *avertis* pour ce qui concerne le choix de fonctionnalités un peu plus spécifiques.

2.3 Définition des objectifs.

Les objectifs principaux du projet peuvent être résumés par le diagramme des cas d'utilisation suivant :

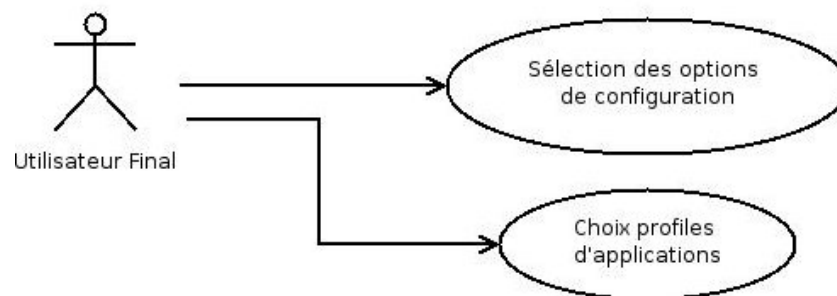


FIG. 1 – Diagramme UML – Cas d'utilisation du projet OS A La Carte.

Pour pouvoir profiter du projet Os A La Carte, il suffit à l'utilisateur de se connecter sur le serveur web de notre projet (en l'occurrence hébergé chez sourceforge) et d'effectuer sa configuration personnelle sur une interface web simplifiée.

Une fois la configuration terminée, il aura besoin de **télécharger** et d'**installer** sa version personnalisée de Linux.

3 Contexte.

3.1 Étude de l'existant.

Notre projet utilisera beaucoup d'outils déjà **existants**. La réutilisation de programme est possible car ceux-ci sont disponibles sous *licence Open Source* (GPL), nous disposons donc des sources des programmes et avons le droit de les modifier et de les redistribuer. Ainsi, en cas de problème nous pouvons donc nous même corriger les bogues et ne sommes donc pas obligé d'attendre le bon vouloir d'un éditeur de logiciels.

La distribution Linux Ubuntu, gère les applications à l'aide de paquets. Ces paquets sont regroupés dans des dépôts. Actuellement Ubuntu gère 22 000 paquets.

D'après la documentation qui est disponible en Annexe sous le nom « *Documentation Custom Install CD* », nous avons mis en évidence le fonctionnement du processus d'installation.

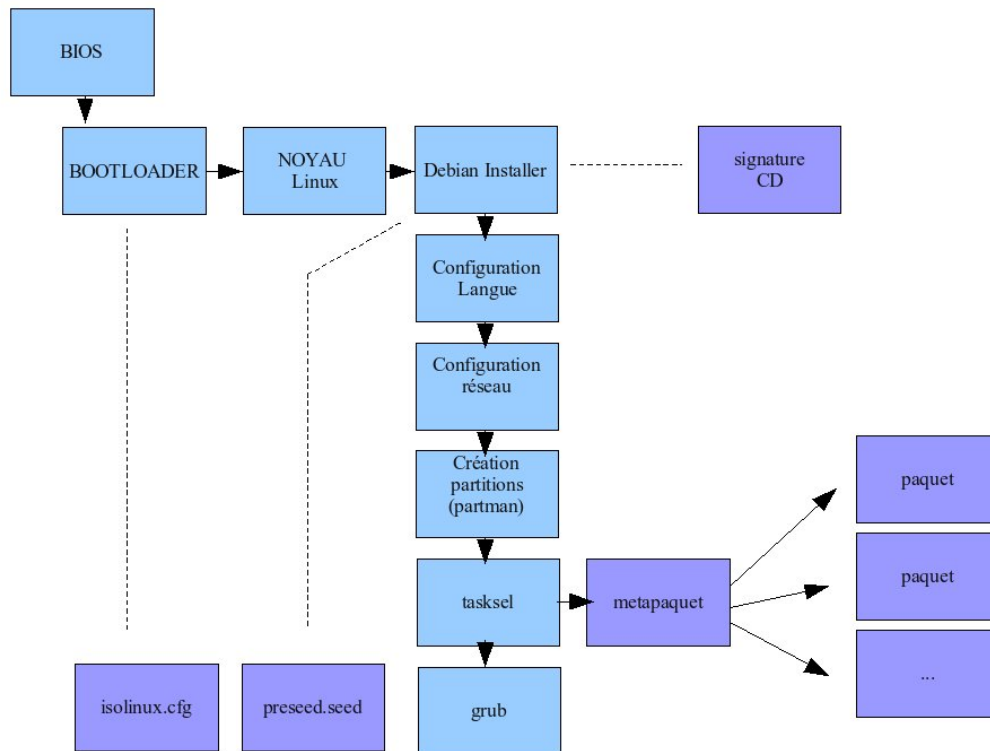


FIG. 2 – Schéma – Différentes étapes du procédé d'installation.

Note : Notre objectif sera la modification du DVD d'installation, nous modifierons ici principalement les éléments en violet.

3.2 Étude des concurrents.

Il existe beaucoup de systèmes d'exploitation Linux, comme décrit en présentation du projet (Partie Origine), leur utilisation repose sur le même modèle.

En effet, l'utilisateur **télécharge** l'image du CD ou DVD d'installation, la grave, puis lance le processus d'installation. L'utilisateur suit donc les différents menus qui lui sont proposés, une fois l'installation terminée, il va ensuite configurer ses périphériques, installer de nouveaux programmes et enfin configurer ses applications selon ses préférences.

Ainsi, notre projet est **novateur**, car il n'existe pas de solution qui configure et personnalise le système d'exploitation depuis un page internet et ainsi arrivé à un produit totalement personnalisé.

3.3 Environnement technique.

Notre projet sera développé sous le système d’exploitation Ubuntu Linux « *Edgy Eft* ».

Nous utiliserons comme Environnement de Développement Intégré Eclipse avec ses modes **Python** (Pydev) et **PHP**.

Les codes sources de notre projet seront hébergés sur **SourceForge**. (CF <http://osalcproject.sourceforge.net/>)

Pour développer le projet, nous utiliserons le langage orienté objet **Python**, car c’est un langage évolué qui se rapproche le plus des scripts **Bash** (scripts qui sont fournis avec la documentation – « CF Annexe Documentation Officielle : Custom Install CD »).

En ce qui concerne l’interface web avec l’utilisateur, nous utiliserons le langage *PHP* dans sa version **5.1**. Pour manipuler les paquets et stocker les informations relatives aux paquets nous utiliserons une base de données **Mysql** dans sa version 5.

En Environnement de test (pour les tests d’intégration), nous utiliserons la virtualisation de système d’exploitation avec des outils comme **VMWare** (propriétaire mais gratuit) ou alors **Qemu**.

3.4 Ressources mises à disposition.

Nous utiliserons nos ordinateurs personnels pour développer le projet.

L’hébergement sur **SourceForge**, nous permettra d’utiliser un dépôt commun pour mettre le code source, ce qui nous permettra de synchroniser le code entre les différents développeurs.

Nous utiliserons les outils de **rapports de bugs** avec les clients et finalement les **mailings lists** si des utilisateurs nécessitent par la suite des fonctionnalités très spécifiques. (CF Annexe : Dossier Hébergement du projet SourceForge)

3.5 Respect d’un règlement.

Comme nous utilisons des outils disponible en code ouvert et un hébergeur gratuit nous nous devons de publier nos travaux sous une licence de **logiciel libre**, c’est pour cela que nous avons choisi de publier nos travaux sous la licence **GPL**.

3.6 Suites prévues.

Notre projet pourra intégrer comme évolution, une distinction dans les **profils matériels**, avec notamment des ordinateurs portables (« Laptop ») ou de bureau (« Desktop »). Selon le choix de l’utilisateur, la configuration du système d’exploitation sera différente, et des applications supplémentaires seront installées (gestion de l’économie d’énergie, de programme d’utilisation du réseau WIFI, ...).

L'analyse préliminaire, que l'on peut trouver dans le *Schéma – Découpage du projet OS A La Carte en modules.*, fait état d'un module qui s'appelle « Schedule » ou ordonnancement en français.

Si nous avons le temps nous essayerons de gérer une file d'attente des utilisateurs, le procédé de personnalisation prenant un certains temps (environ 3 minutes sur une machine récente), il sera important de notifier à l'utilisateur son temps d'attente.

De plus, le projet restera **ouvert** à d'autres modifications concernant des programmes qui ne sont pas présents dans l'analyse actuelle.

4 Directives particulières.

4.1 Contraintes de développement.

Dans le but de résoudre les points techniques critiques, nous avons réaliser un **prototype** du programme, qui était l'application de la documentation officielle (Cf Annexe – Documentation Officielle : Custom Install CD).

Nous envisageons une méthode **Agile** pour le développement tout en tenant compte du fait qu'il ne faudra pas négliger l'élaboration des dossiers de spécifications et de conceptions et des protocoles de tests.

Nous envisageons la méthode « *Extreme programming* » qui fonctionne généralement par **binôme**. Nos tests seront fait avant chaque implémentation. De plus cette méthode permet de dépanner un développeur quand celui est bloqué, en effet son collègue vient lui prêter assistance.

4.2 Contraintes méthodologiques.

Nous utiliserons **Merise** (Modèle Conceptuel des Données) et **UML** (Diagramme de Classe) pour modéliser les données et UML pour les enchaînements ou relations entre les différents modules.

4.3 Contraintes de sécurité.

Notre travail repose sur la distribution Linux Ubuntu. Cette distribution repose sur un cycle de développement de **6** mois. Ainsi, tous les semestres, une nouvelle version apparaît, en général Avril et Octobre.

Le numéro de version d'Ubuntu est composé du numéro de l'année suivi du mois, ainsi pour la version que nous étudierons, appelée « Edgy Eft », qui est sortie en Octobre 2006 sa version est **6.10**.

Voici un historique des versions d'Ubuntu qui sont sorties depuis sa création :

Version	Date de sortie	Nom de Code	Traduction nom de code
Ubuntu 4.10	20 Octobre 2004	« Warty Warthog »	« Le phacochère verruqueux »
Ubuntu 5.04	8 Avril 2005	« Hoary Hedgehog »	« Le hérisson vénérable »
Ubuntu 5.10	13 Octobre 2005	« Breezy Badger »	« Le blaireau jovial »
Ubuntu 6.06 *	1 ^{er} Juin 2006	« Dapper Drake »	« Le canard pimpant »
Ubuntu 6.10	26 Octobre 2006	« Edgy Eft »	« Le triton nerveux »
Ubuntu 7.04	19 Avril 2007	« Feisty Fawn »	« Le faon téméraire »

FIG. 3 – Tableau – Historique des sorties des différentes versions d'Ubuntu.

* Note : Cette version est sortie plus tard, en Juin au lieu d'Avril, car cette version a une durée de support plus importante que les autres (elle est appelée, LTS pour *Long Term Support*).

Cependant entre chaque sortie de version, il y a des **mise à jour de sécurité**. Notre projet qui devra vraisemblablement sortir entre deux versions finales, au plutard en février 2007, devra intégrer les **mise à jour critiques** pour ne pas diffuser un produit qui contienne des failles de sécurité.

Depuis la sortie d'Ubuntu « Edgy Eft », on dénombre la sortie de **6** alertes de sécurité :

Numéro	Date	Titre	Paquet(s) Affecté(s)
USN-387-1	28/11/2006	dovecot vulnerability	dovecot-common-1.0.rc2-1ubuntu2.1
USN-380-1	11/11/2006	avahi vulnerability	avahi-daemon-0.6.13-2ubuntu2.2
USN-372-1	01/11/2006	imagemagick vulnerability	libmagick9-6.2.4.50.10ubuntu0.1
USN-369-2	02/11/2006	PHP vulnerability	libapache2-mod-php5-5.1.6-1ubuntu2.1
USN-369-2	01/11/2006	postgresql-8.1 vulnerabilities	postgresql-8.1.4-7ubuntu0.1
USN-370-1	31/10/2006	screen vulnerability	screen-4.0.2-4.1ubuntu5.6.10

FIG. 4 – Tableau – Alertes de sécurité d'Ubuntu pour sa version « Edgy Eft ».

Il est donc important d'être synchroniser avec les dépôts d'ubuntu et d'intégrer les mise à jour de sécurité dans l'image DVD que nous distribuerons au client.

5 Expression fonctionnelle du besoin.

Parmi tous les niveaux d'abstraction qu'il existe dans un système d'exploitation, notre travail portera sur la personnalisation du niveau le plus élevé principalement, à savoir au dessus du serveur graphique. Notre domaine d'action est représenté ici en violet :

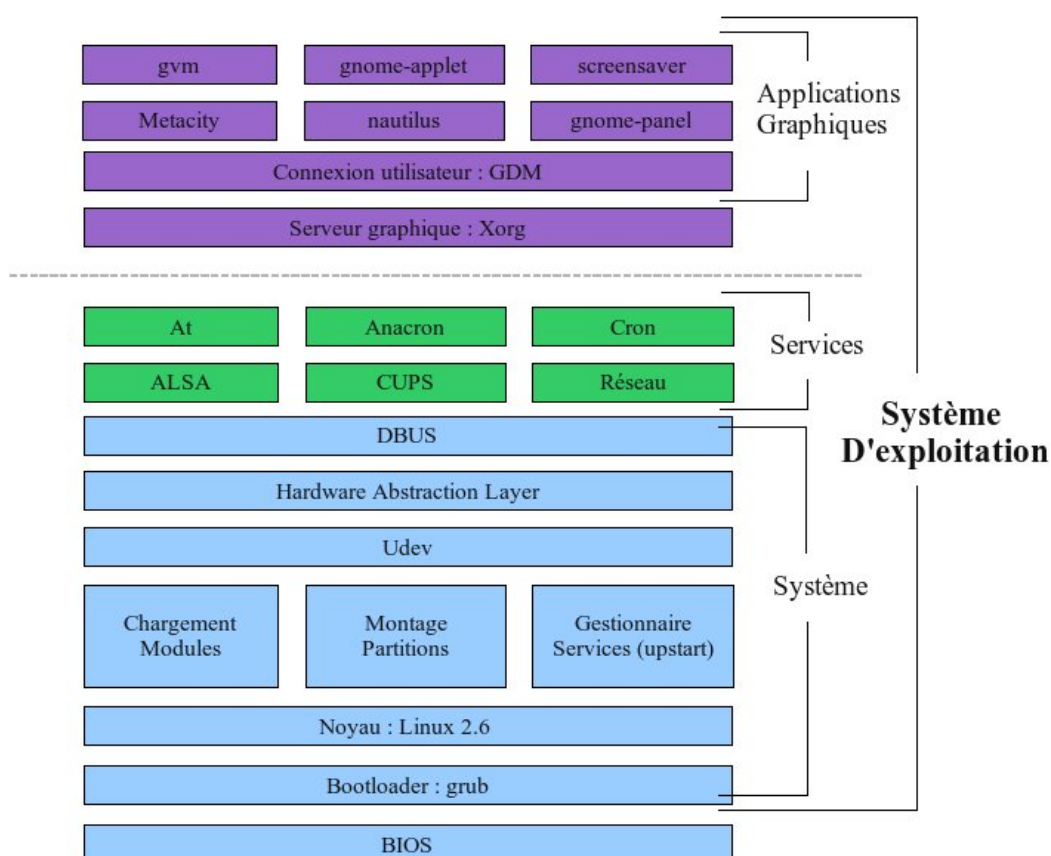


FIG. 5 – Schéma – Domaine d'intervention du projet OS A La Carte.

Notre projet se divisera en **3** parties : *paramétrage du CD d'installation*, *modification de paquets existants* et *enfin l'installation d'applications additionnelles*.

6 Exigences qualité.

Pour la gestion qualité de notre projet, nous utiliserons les normes suivantes :

- La norme ISO/IEC 12207 : qui est une norme générique relative aux processus de cycles de vie en général, et donc méthode de l'Extreme Programming utilisé pour le développement de notre projet.

- La norme Afnor NF-X50-151 : décrivant la méthode de l'analyse fonctionnelle et de sa documentation.

7 Aspect financier.

Ce dossier se fera vers la fin du projet et portera certainement sur la simulation de la **mise en production** du service, à savoir la vente des Systèmes d'exploitation et la mise en place des machines qui seront utilisées pour la création du DVD.

8 Remise de la documentation.

Les documents de suivi (le journal de bord) de notre projet « OS A La Carte » devront être remis à M Alexandre *LEPRIEULT*, le tuteur de notre projet, au format papier et dans les délais indiqué dans le planning prévisionnel. Ces documents sont :

Le projet sera constitué des documents suivants :

- Cahier des charges ;
- Cahier de spécifications logicielles ;
- Dossier de conception ;
- Journal de bord (constitué du compte de la réunion précédente, et de l'état d'avancement du projet) ;
- Dossier final présentant la conduite du projet ;
- Protocoles de tests.

9 Planning prévisionnel.

Avant d'effectuer le planning prévisionnel, et de répartir les tâches, il convient de réaliser une analyse rapide du projet et de découper en modules les différents aspects de celui-ci.

Une analyse fait apparaître que le projet sera divisé en 8 modules qui sont :

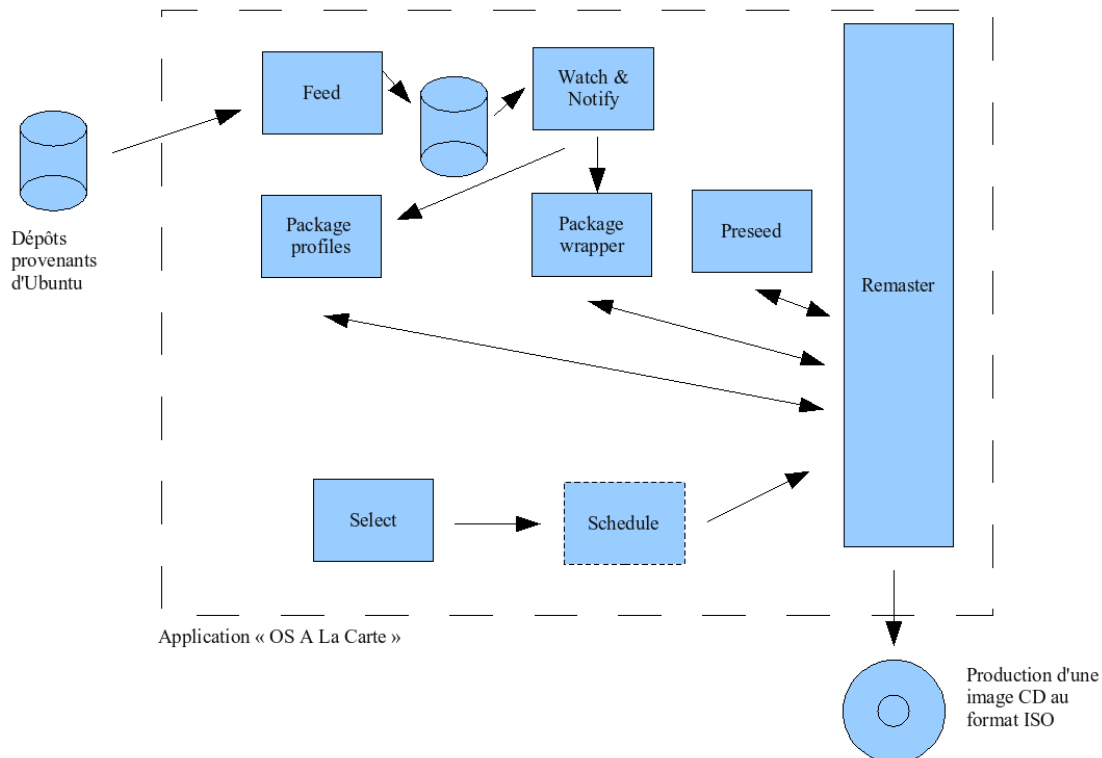


FIG. 6 – Schéma – Découpage du projet OS A La Carte en modules.

Rôle des modules :

- « **Feed** » prendra en compte le suivi des mises à jour proposées par les dépôts officiels d'Ubuntu (Réponse à la contrainte de *sécurité*) ;
- « **Watch and Notify** » aura pour rôle d'afficher les changements des dépôts ;
- « **Package Profiles** » s'occupera de la gestion des profils d'applications ;
- « **Package Wrapper** » gèrera les paquets, à savoir leur ouverture, leur modification, et finalement leur reconstruction ;
- « **Preseed** » concernera le paramétrage du CD d'installation ;
- « **Remaster** » aura pour tâche la création de l'image ISO en prenant en compte les modules « **Package Profile** », « **Package Wrapper** » et « **Preseed** » ;
- « **Select** » sera l'interface où l'utilisateur sélectionnera ses choix ;
- « **Schedule** » gèrera les utilisateurs, avec une file d'attente, la sauvegarde des sélections des utilisateurs (Module Non Prioritaire).

10 Maintenance.

« OS A La Carte » est un projet **libre** sous licence GPL, et sera publié sur **SourceForge**. Il sera donc de notre responsabilité directe de corriger les bugs qui peuvent être mis en évidence par les utilisateurs de l'application.

Comme il s'agit d'un projet diffusé sous licence Logiciel Libre (GPL), toute personne qui trouvera le projet intéressant pourra participer à son élaboration ou même à sa reprise, si par la suite, nous décidons de l'abandonner pour nous consacrer à autre chose.

11 Assistance.

En ce qui concerne l'assistance, nous utiliserons un système de rapport de bogues (bugtracking system), qui sera disponible sur le site de SourceForge (CF <http://osalcproject.sourceforge.net>), pour permettre aux utilisateurs de nos services de demander de nouvelles fonctionnalités ou même signaler des bogues.

Ainsi, grâce à ce système de suivi des bogues, les utilisateurs sont en contact continu avec les développeurs du projet pour, en fin de compte, participer à l'amélioration du projet.

ISAIP

18, rue du 8 Mai 1945

49000 St Barthélémy d'Anjou



Dossier de Spécifications Logicielles

Développement d'une application de personnalisation de système
d'exploitation : "OS A La Carte"

17 Janvier 2007

Mohammed BADISS, Jérôme CHARLOT

2^{ème} année CPI

Promotion 2005 – 2007

Sous la responsabilité de :
M Alexandre LEPRIEULT ;



Table des matières

1	Traçabilité	3
2	Introduction	3
2.1	But du document	3
2.2	Contexte de l’application	3
2.3	Documentation	3
2.3.1	Documents applicables	3
2.3.2	Documents de référence	3
3	Spécifications générales	3
3.1	Présentation du logiciel	3
3.1.1	Environnement du logiciel	3
3.1.2	Architecture logicielle	4
3.1.3	Description générale du logiciel	5
3.2	Contraintes structurelles	6
3.3	Contraintes opérationnelles	8
3.3.1	Performances	8
3.3.2	Installation	9
4	Spécifications fonctionnelles	10
4.1	Vue d’ensemble	10
4.2	Fonctionnalités au niveau du DVD d’installation	11
4.3	Modifications des <u>paquets</u> les plus importants.	12
4.3.1	Le navigateur Internet (Firefox)	12
4.3.2	L’éditeur de texte (Gedit)	12
4.3.3	Le terminal (Gnome-terminal)	13
4.3.4	Le bureau (Nautilus/Metacity)	13
4.3.5	L’explorateur de fichiers (Nautilus)	13
4.3.6	Les bureaux virtuels (Metacity)	13
4.3.7	L’interpréteur de commandes (Bash)	14
4.4	Choix des profils d’applications	15
4.4.1	Référencement de tous les profils d’applications	15
4.4.2	Profile “Archive”	15
4.4.3	Profile “Codecs Vidéos et Audio”	16
4.4.4	Profile “Composition de documents Latex”	16
4.4.5	Profile “Environnement de Développement Intégré”	16
4.4.6	Profile “Développement Java”	17
4.4.7	Profile “Développement Python”	18
4.4.8	Profile “Sécurité Réseau”	18
4.4.9	Profile “Développement Web”	19
4.4.10	Profile “Retouche d’images”	19
4.4.11	Profile “Édition Vidéo”	20
4.4.12	Profile “Compatibilité Windows”	20
4.4.13	Profile “Outils téléchargement”	20
4.4.14	Profile “Lecteur video secondaire”	21
4.4.15	Profile “SoftPhones”	21
4.4.16	Profile “Pack fontes”	21

5	Spécifications d’interface	22
5.1	Découpage de l’application en modules	22
5.2	Interfaces Homme – Machine	24
5.3	Interfaces Logiciel – Logiciel	25

Table des figures

1	Schéma – Architecture logicielle de l’application OS A La Carte.	4
2	Tableau – Composants logiciels de l’application OS A La Carte.	5
3	Schéma – Architecture matérielle pour le déploiement de l’application Os A La Carte. .	7
4	Schéma – Échanges d’informations concernant l’état entre le répartiteur de charge et les différents serveurs.	8
5	Tableau – Estimation du temps de téléchargement du DVD d’installation selon les diffé- rents types de connexion.	9
6	Schéma – Choix possibles de l’utilisateur.	10
7	Tableau – Récapitulatif de l’occupation des profils d’applications.	15
8	Schéma – Découpage de l’application OS A La Carte en modules.	22
9	Tableau – Récapitulatif des modules et de leur type d’accès à la base de données.	23
10	IHM – Interface non détaillée de sélection de configuration de l’utilisateur	24
11	IHM – Interface détaillée de sélection de configuration de l’utilisateur	25
12	Schéma – Communication entre les modules “Select” et “Feed” à l’aide d’un identifiant.	26

1 Traçabilité

Version	Date	Commentaires
1.00	15/12/06	Première version
1.10	25/12/06	Ajout des dépendances pour la partie "Package Profile"
1.20	13/01/07	Correction de la partie "Package Profile" (Ajout colonne "Type")
2.00	03/02/07	Ajout de la spécification Homme - Machine
2.10	04/02/07	Correction orthographe et amélioration schéma

2 Introduction

2.1 But du document

Il s'agit là de présenter la phase de spécifications des besoins du logiciel d'un point de vue méthodologique. Il dépend très peu de l'application elle même.

2.2 Contexte de l'application

Ce paragraphe présente l'application dans son contexte.

2.3 Documentation

2.3.1 Documents applicables

Pour la réalisation de ce projet, nous nous basons sur un **tutorial** (CF Annexe A "Custom Install CD") fourni sur le site officiel "Ubuntu". Ce tutorial explique la méthode de modification d'un DVD d'installation dans le but de personnaliser le processus d'installation.

2.3.2 Documents de référence

Le document de référence principal de ce document est le **Cahier des Charges** qui est disponible dans ce dossier.

3 Spécifications générales

3.1 Présentation du logiciel

3.1.1 Environnement du logiciel

Pour pouvoir accéder à notre interface web de sélection, il suffit d'avoir un navigateur web, et une connexion internet **haut débit** type "ADSL" ou "Cable". De plus, notre service est **indépendant** du système d'exploitation que le client utilise (Microsoft Windows, Apple Mac OS ou encore Linux).

Pour pouvoir installer le système d'exploitation, il faut également un graveur de DVD, un logiciel de gravure et un support de stockage (un DVD).

3.1.2 Architecture logicielle

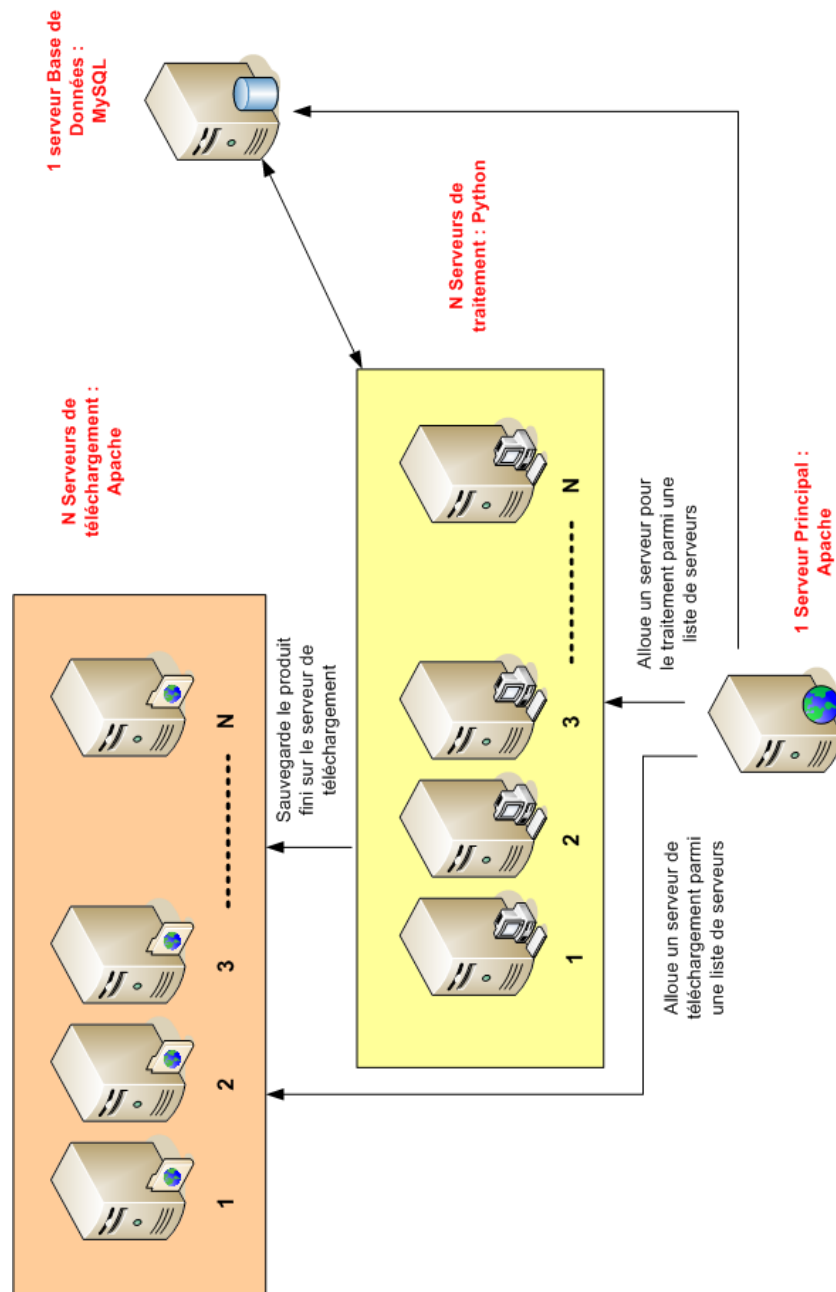


FIG. 1 – Schéma – Architecture logicielle de l'application OS A La Carte.

Ainsi notre architecture fait état de **quatre** types de serveurs qui sont :

1. Un serveur *web*, ou le client réalise sa sélection ;
2. Un ou plusieurs serveurs qui réalisent le *traitement* (génération de l'image du DVD) ;
3. Un serveur de *téléchargement* qui met à disposition l'image au format ISO au client ;
4. Un serveur de *base de données*, qui stocke les informations de l'utilisateur.

Voici les composants logiciels :

Type de Serveur	Programme	Version
Web	Apache	2.0.55
Web	PHP5	5.1.6
Web	php5-mysql	5.1.6
Téléchargement	Apache	2.0.55
Base de données	Mysql	5.0.24
Traitement	Python	2.4.4
Traitement	Python-mysqldb	1.2.1

FIG. 2 – Tableau – Composants logiciels de l'application OS A La Carte.

3.1.3 Description générale du logiciel

Le fonctionnement du logiciel peut être divisé en **deux** étapes principales. Lors de la première étape, l'utilisateur se connecte sur notre interface web pour **personnaliser** le DVD d'installation du système d'exploitation.

Cette interface de sélection offre un certain nombre de **thèmes de personnalisation** (configuration du navigateur internet, configuration des bureaux virtuels, ...), et pour chaque thème elle offre plusieurs options. Et ainsi chaque choix entraîne la modification des paquets concernés.

Une fois que l'utilisateur a choisi sa configuration, notre interface web le **met en attente** le temps de générer son DVD d'installation. La sélection de l'utilisateur est **sauvegardée** sur la base de données. Pour l'instant nous nous contentons d'un ordonnancement de type **FIFO** (premier arrivé, premier servi) pour servir les clients.

Ensuite, la deuxième étape consiste à la **génération du DVD** d'installation personnalisée. Après que l'utilisateur ait validé ses choix, un script **Python** lit sa configuration depuis la base de données, modifie les paquets concernés, et génère un fichier image ISO (environ 1,1 Go). C'est ce fichier qui sera proposé au téléchargement sous forme d'un lien direct dans l'interface web.

3.2 Contraintes structurelles

Avec la mise en production du service de personnalisation de système d’exploitation, il nous faudra prendre en compte l’**évolution** de la demande. Pour cela il nous faudra **distribuer** le procédé de personnalisation selon le schéma suivant :

Ainsi, il est nécessaire de distribuer **deux** activités différentes qui sont le **traitement** et la **mise à disposition** de l’image ISO (ou serveur de téléchargement).

Pour distribuer sur les deux types de machines nous devons tenir compte de leur **état** comme l’occupation de la machine (occupation du processeur) et la disponibilité de la bande passante.

Ces deux critères sont **indépendants**, on répartit d’abord sur une machine de traitement puis sur une autre machine de mise à disposition du fichier. Voici un schéma qui représente les échanges entre le **répartiteur de charge** et les **serveurs** de traitement et de téléchargement :

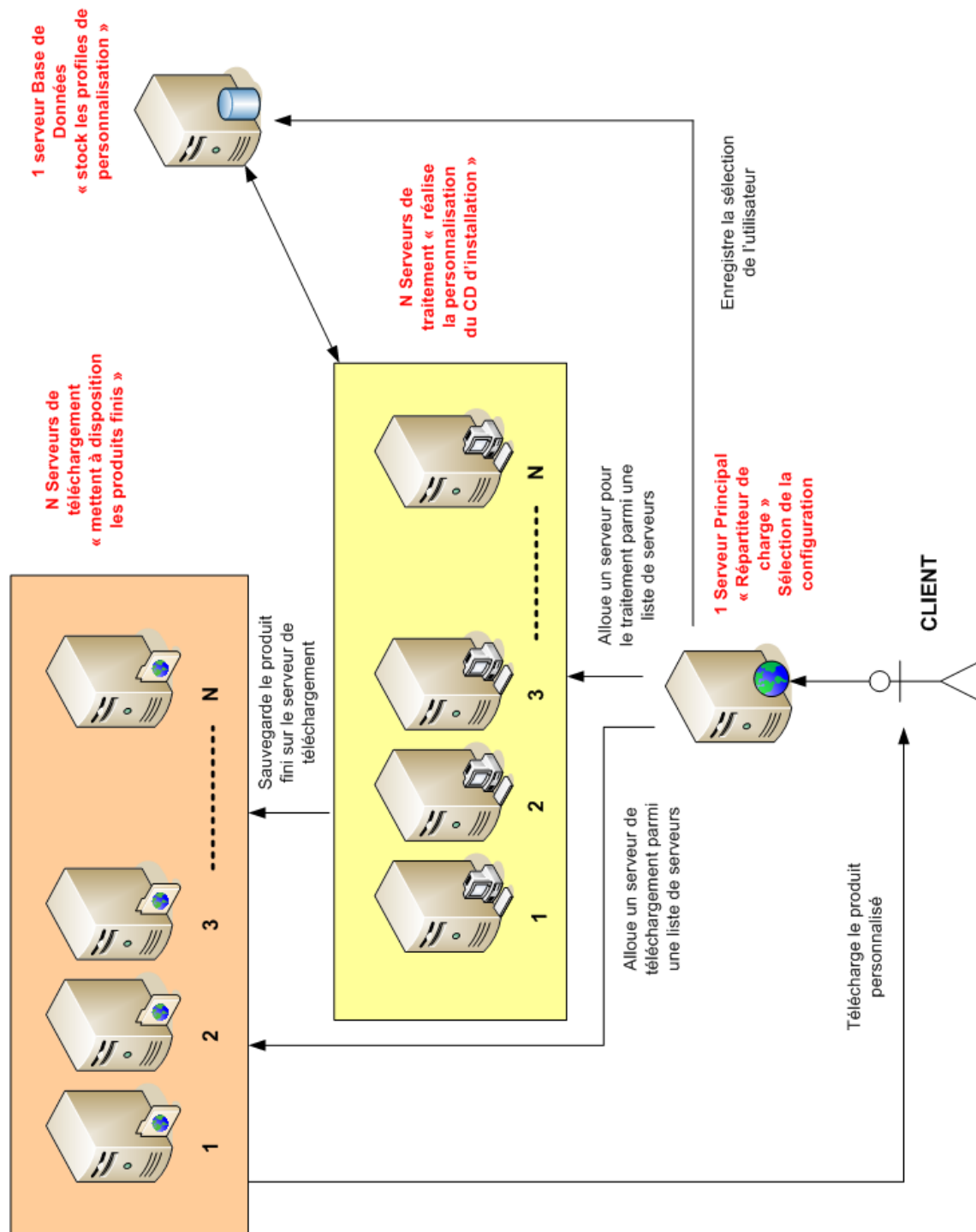


FIG. 3 – Schéma – Architecture matérielle pour le déploiement de l'application Os A La Carte.

Nous disposons donc d'une liste de machines qui sont disponibles, et lorsqu'un client fait la demande de personnalisation, on envoie une requête pour connaître l'état des différents serveurs.

Note : Cette contrainte fait partie du module "Schedule" et est définie comme **optionnelle**, elle sera

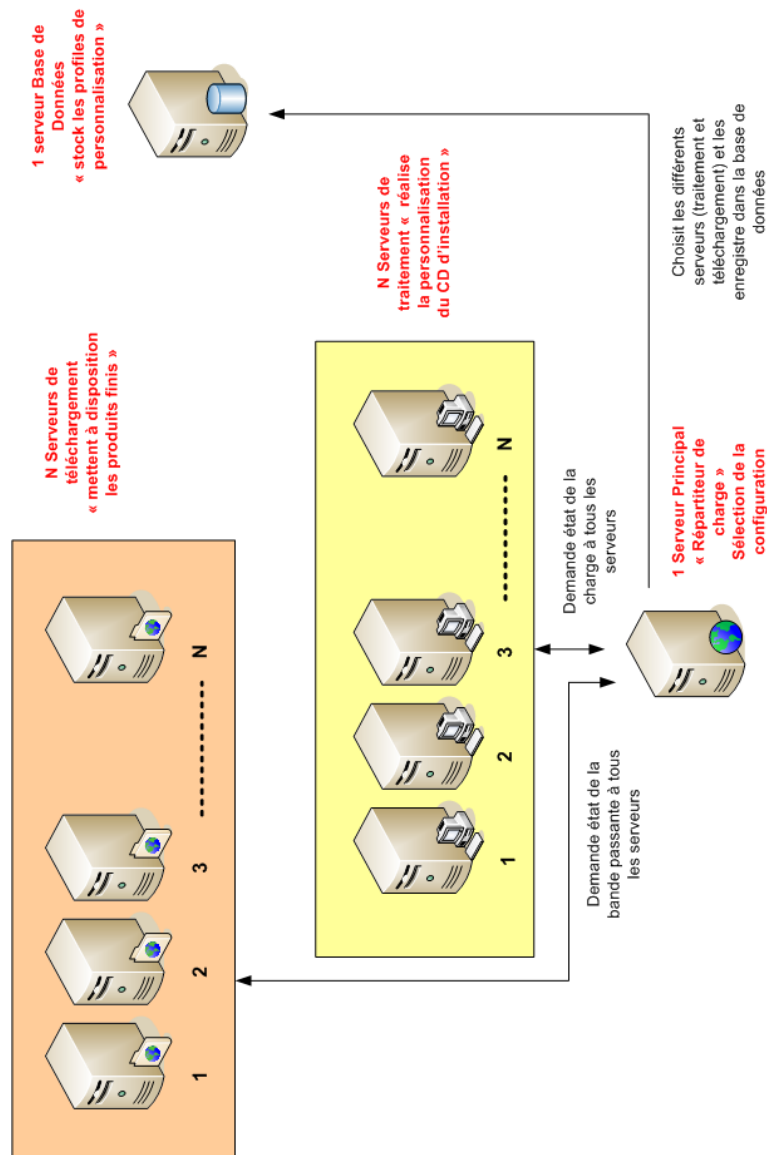


FIG. 4 – Schéma – Échanges d'informations concernant l'état entre le répartiteur de charge et les différents serveurs.

donc réalisée à la fin. Cependant il est important de la définir dans le dossier de spécifications dans le but de **prévoir** le codage de l'application pour éviter d'avoir des changements trop conséquents à faire à la fin du projet.

3.3 Contraintes opérationnelles

3.3.1 Performances

Nous allons distinguer **trois** phases pour la présentation des performances requises :

1. La première phase est la **sélection** de la configuration depuis notre interface Web. Le temps de

réponse l'interface web dépend principalement de la bande passante du client, et de notre serveur. Mais avec les connexions actuelles, la technologie ADSL étant largement répandue, ce temps est **négligeable** (à peine quelques secondes). Cependant cette phase peut être influencée également par le nombre d'utilisateurs.

2. Une fois la phase de personnalisation terminée. La phase de **génération** du DVD d'installation peut exiger environ **15** minutes sur le serveur dédié (un Pentium 4 ou équivalent).
3. Finalement, la phase de **téléchargement** de l'image du DVD est la plus **exigeante** au niveau temps, car comme la première phase, elle dépend de la bande passante que dispose le client mais surtout celle que nous allouons pour le serveur de téléchargement. Ainsi dans des conditions idéales, à savoir si nos serveurs peuvent suivre la charge de la bande passante (que nous disposons de suffisamment de bande passante), on peut dresser un tableau du temps d'attente nécessaire à l'utilisateur pour télécharger l'image du DVD d'installation :

Type de connexion	Débit effectif descendant	Temps de téléchargement
1 Mega	130 Ko/s	5h12min
8 Mega	1 Mo/s	18min
20 Mega	2,5 Mo/s	8min

FIG. 5 – Tableau – Estimation du temps de téléchargement du DVD d'installation selon les différents types de connexion.

Note : On suppose que l'utilisateur décide de sélectionner **tous** les profils d'applications qui sont disponibles soit un ajout d'environ 350 Mo au DVD qui fait 700 Mo nous arrivons donc à un total de **1,1 Go** à télécharger.

Ainsi le temps d'attente total pour l'utilisateur ayant une très bonne connexion sera de **23min** alors que pour un utilisateur ayant une connexion dite "moyenne" sera de **5h28min**.

3.3.2 Installation

L'utilisateur télécharge une image de DVD d'installation **au format ISO**, et il est à sa charge de la graver sur un DVD avant de l'installer. Mais nous prévoyons, prochainement, graver les DVD nous même et les **vendre** aux clients.

4 Spécifications fonctionnelles

4.1 Vue d'ensemble

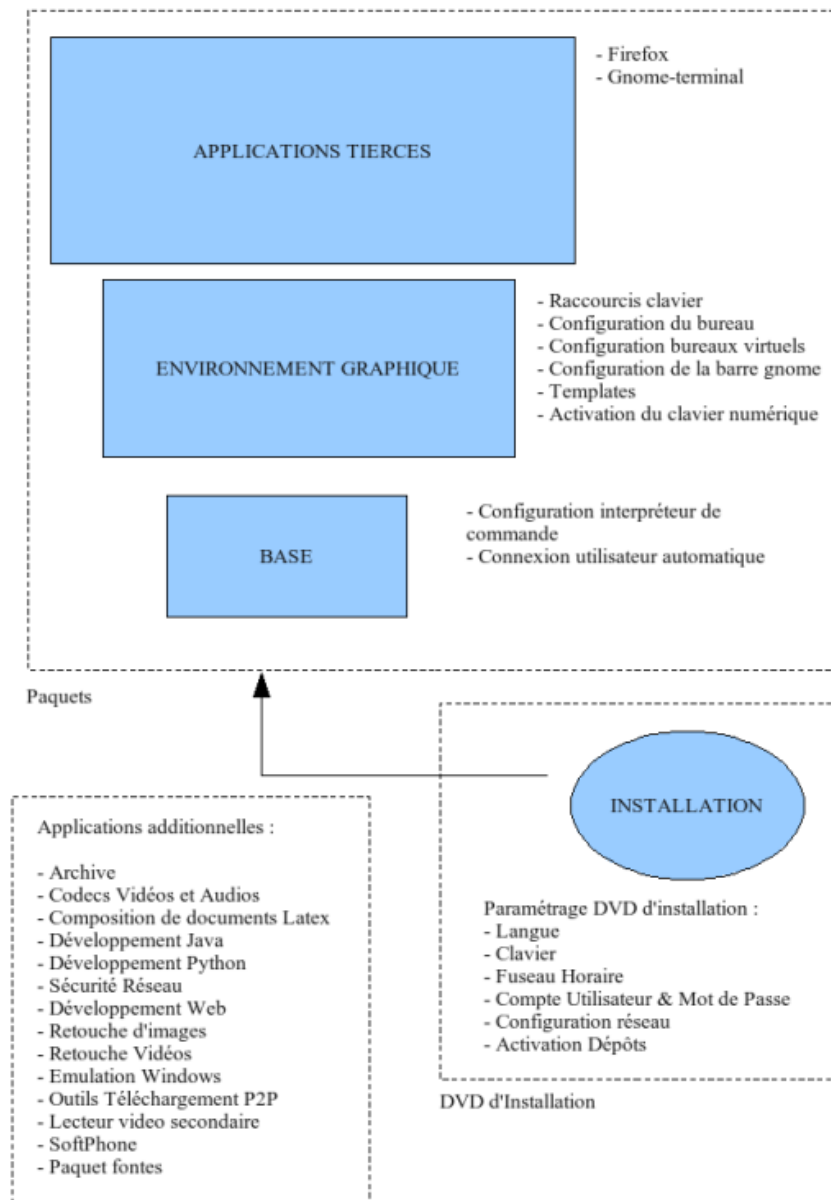


FIG. 6 – Schéma – Choix possibles de l'utilisateur.

L'utilisateur personnalise le DVD d'installation à **trois** niveaux différents.

Les trois niveaux sont :

1. Fonctionnalités au niveau du DVD d'installation.
2. Modifications des paquets les plus importants.

3. Choix des profils d'applications.

Ces différents niveaux sont **abstraits** et **transparents** pour l'utilisateur, ils correspondent à un découpage en modules de l'application. Ce découpage est détaillé dans la partie “*Découpage de l'application en modules*”.

4.2 Fonctionnalités au niveau du DVD d'installation

Cette étape concerne la modification du DVD d'installation, en effet le programme d'installation (ici “debian-installer”) a été conçu dans le but de prendre en compte un certains nombres de paramètres qui sont liés au **coeur** du fonctionnement du système d'exploitation. Ces paramètres sont transmis via un **fichier texte** qui possède une structure spéciale.

Ainsi à ce niveau, l'utilisateur peut choisir parmi les options suivantes :

- Choix de la langue par défaut (parmis une liste de choix) ;
- Configuration du réseau * ;
- Nom de la machine sur le réseau ;
- Nom du domaine ;
- Choix également son fuseau horaire ;
- Définition du compte utilisateur et de son mot de passe ;
- Activation des dépôts additionnels **.

* Cette étape sera détaillée et on aura le choix entre une configuration en **automatique** (DHCP) ou une configuration **manuelle**. Dans le cas ou l'utilisateur souhaite une configuration manuelle du réseau, il devra spécifier les informations suivantes :

- Configuration de l'adresse logique (IP) de la machine ;
- Configuration de l'adresse IP du serveur de noms (DNS) ;
- Configuration du masque réseau (Netmask) ;
- Configuration de l'adresse IP de la passerelle.

** Cette sélection permet à l'utilisateur de pouvoir installer plusieurs autres paquets qui ne sont pas activés par défaut pour des raisons de respect de licence d'utilisation et de brevet.

4.3 Modifications des paquets les plus importants.

Le deuxième type de personnalisation se situe au niveau des paquets eux même. Nous avons regroupé les modifications par thèmes.

4.3.1 Le navigateur Internet (Firefox)

L'utilisateur pourra alors activer ou non les options suivantes :

1. Utiliser l'ancien mode de fermeture des onglets (une croix à droite en haut de l'application au lieu d'avoir une croix par onglet) ;
2. Cacher le bouton "Go" qui permet de lancer l'adresse saisie dans la barre d'adresse ;
3. Forcer l'ouverture des liens "externes" dans la même page dans un nouvel onglet ;
4. Désactiver la demande de fermeture d'onglets lorsque l'on quitte l'application ;
5. Activer le bouton du milieu ;
6. Activer le défilement avec le bouton du milieu ;
7. Désactiver le préchargement des sites (pour accélérer la navigation) ;
8. Désactiver le cache sur le disque (pour les machines peu performantes) ;
9. Désactiver l'historique de navigation (pour les machines peu performantes) ;
10. Désactiver les cookies (Sécurité) ;
11. Activer les cookies uniquement pour la session en cours d'utilisation (Sécurité) ;
12. Désactiver l'historique des téléchargements effectués.

Avec le navigateur Internet on peut également activer les plugins suivants :

- Plugin Java (permet d'exécuter des programmes Java) ;
- Plugin Flash (permet de lire des animations au format Flash) ;
- Plugin Vidéo (permet de lire les vidéos dans le navigateur) ;

4.3.2 L'éditeur de texte (Gedit)

On se propose de modifier les paramètres suivants :

1. Activer l'indentation automatique ;
2. Surligner la ligne en cours de modification ;
3. Taille de la police par défaut (10) ;
4. Afficher les numéros de lignes ;
5. Désactiver les sauvegardes automatiques (fichiers commençant par un "tilde") ;
6. Désactiver les copies de sauvegarde ;
7. Insérer des espaces plutôt que des tabulations pour l'indentation ;
8. Liste des plugins activés par défaut : [docinfo,indent,modelines,pythonconsole,savecopy,spell,time].

4.3.3 Le terminal (Gnome-terminal)

On se propose de modifier les paramètres suivants :

1. Activer les raccourcis clavier ;
2. "CTRL + Q" pour fermer l'onglet du terminal ;
3. "CTRL + T" pour ouvrir un nouvel onglet ;
4. "CTRL + Flèche de droite" pour aller à l'onglet placé à droite ;
5. "CTRL + Flèche de gauche" pour aller à l'onglet placé à gauche.

4.3.4 Le bureau (Nautilus/Metacity)

On se propose de modifier les paramètres suivants :

1. Activer l'icône "Poste de Travail" sur le Bureau ;
2. Activer l'icône "Dossier Personnel" sur le Bureau ;
3. Activer l'icône "Corbeille" sur le Bureau ;
4. Activer l'icône "Parcourir le Réseau" sur le Bureau ;
5. Activer les icônes de type "Points de Montage" (clés USB, partitions, ...) sur le Bureau ;
6. Activer les raccourcis claviers ;
7. "Touche Windows + e" pour lancer l'explorateur de fichiers ;
8. "Touche Windows + g" pour lancer l'éditeur de texte ;
9. "Touche Windows + o" pour lancer le traitement de texte ;
10. "Touche Windows + n" pour lancer le navigateur internet ;
11. "Touche Windows + m" pour lancer le logiciel de messagerie ;
12. "Touche Windows + c" pour lancer le terminal.

4.3.5 L'explorateur de fichiers (Nautilus)

On se propose de modifier les paramètres suivants :

1. Utiliser un agencement compact (utile pour les portables) ;
2. Activer les scripts de modifications d'images ;
3. Activer les modèles de documents (traitement de texte, tableur, ...).

4.3.6 Les bureaux virtuels (Metacity)

On se propose de modifier les paramètres suivants :

1. Définir le nombre de bureaux virtuels que l'utilisateur souhaite utiliser (de 1 à 8) ;
2. Définir leur nom, par exemple "Communication", "Multimédia", "Internet", "Travail".

4.3.7 L'interpréteur de commandes (Bash)

On se propose de modifier les paramètres suivants :

1. Ajouter la complétion aux commandes saisies par l'utilisateur ;
2. Coloration du “prompt” ;
3. Ajout d'alias (une liste maximale de 10 alias) .

Ensuite pour terminer, on propose d'installer des profils d'applications. Chaque profile regroupe un nombre de programmes. Les profils d'applications sont détaillés dans la partie suivante.

4.4 Choix des profils d'applications

4.4.1 Référencement de tous les profils d'applications

Nous avons donc regroupé les applications selon leur utilisation.

Voici un tableau récapitulatif :

Nom du profil	Nombre de paquets	Taille sur le DVD
alc-archive	5	1 Mo
alc-codecs-xine	5	19 Mo
alc-latex	4	38,8 Mo
alc-edition	7	86,9 Mo
alc-edition-java	5	88,6 Mo
alc-edition-python	5	4 Mo
alc-network-tools	4	5 Mo
alc-web-development	17	39 Mo
alc-graphic-tools	3	9 Mo
alc-video-edition	1	1,1 Mo
alc-windows-api	1	9,2 Mo
alc-download-soft	2	2,7 Mo
alc-videoplayer	2	3,5 Mo
alc-softphones	7	14 Mo
alc-fonts	1	23 Ko

FIG. 7 – Tableau – Récapitulatif de l'occupation des profils d'applications.

Ainsi si l'utilisateur choisit d'installer tous les profils d'applications proposés, il faudra ajouter **350 Mo** au DVD d'installation ce qui portera la taille finale du DVD à **1,1 Go**.

4.4.2 Profil “Archive”

Référence : alc-archive

Définition : Liste d'applications permettant d'extraire des archives de différents formats.

Taille sur le DVD : 1 Mo

Taille installé : 1 Mo

Contenu :

Nom	Version	Taille installé	Taille Paquet	Licence	Type
rar	3.6.0	255 Ko	255 Ko	GPL	P
unrar-free	0.0.1 (cvs20060609)	18 Ko	18 Ko	GPL	P
unace	1.2b	15 Ko	15 Ko	GPL	P
unace2	2.20	470 Ko	179 Ko	NON FREE	P
p7zip	4.42	470 Ko	316 Ko	GPL	P

Note : Concernant la colonne “Type”, le **P** indique un programme et le **B** une bibliothèque, généralement une dépendance.

4.4.3 Profile "Codecs Vidéos et Audio"

Référence : alc-codecs-xine

Définition : Liste d'applications permettant de lire des vidéos et sons en différents formats (dont DVD).

Taille sur le DVD : 19 Mo

Taille installé : 40 Mo

Contenu :

Nom	Version	Taille installé	Taille Paquet	Licence	Type
w32codecs	20061022	33 Mo	14 Mo	NON FREE	P
libxine-extracodecs	1.1.2	2,6 Mo	1,1 Mo	GPL	B
totem-xine	2.16.2	2 Mo	1,1 Mo	GPL	P
libxine1	1.1.2	3 Mo	2,9 Mo	GPL	B
libdvdcss2	1.2.9	35 Ko	100 Ko	NON FREE	B
libmad0	0.15	75 Ko	75 Ko	GPL	B
libmodplug0c2	3a0.7	112 Ko	112 Ko	GPL	B
libxvnc1	3a1.0.2	12 Ko	12 Ko	GPL	B

Note : Concernant la colonne "Type", le **P** indique un programme et le **B** une bibliothèque, généralement une dépendance.

4.4.4 Profile "Composition de documents Latex"

Référence : alc-latex

Définition : Langage de composition de documents.

Taille sur le DVD : 42,2 Mo

Taille installé : 42,2 Mo

Contenu :

Nom	Version	Taille installé	Taille Paquet	Licence	Type
tetex-bin	3.0-17	3,5 Mo	3,4 Mo	GPL	P
tetex-base	3.0-19	22 Mo	22 Mo	GPL	P
tetex-extra	3.0-19	11 Mo	11 Mo	GPL	P
latex-beamer	3.06-1	2,4 Mo	2,4 Mo	GPL	P
latex-xcolor	2.00	842 Ko	842 Ko	GPL	P
libt1-5	5.1.0	137 Ko	137 Ko	GPL	B
pgf	1.00	2,1 Mo	2,1 Mo	GPL	P
tex-common	0.25	283 Ko	283 Ko	GPL	P

Note : Concernant la colonne "Type", le **P** indique un programme et le **B** une bibliothèque, généralement une dépendance.

4.4.5 Profile "Environnement de Développement Intégré"

Référence : alc-edi

Définition : Environnement de Développement Intégré.

Taille sur le DVD : 98 Mo

Taille installé : 98 Mo

Contenu :

Nom	Version	Taille installé	Taille Paquet	Licence	Type
eclipse-sdk	3.2.1	23 Ko	23 Ko	GPL	P
eclipse	3.2.1	122 Ko	122 Ko	GPL	P
eclipse-jdt	3.2.1	20 Mo	20 Mo	GPL	P
eclipse-pde	3.2.1	6,3 Mo	6,3 Mo	GPL	P
eclipse-platform	3.2.1	29 Mo	29 Mo	GPL	P
eclipse-rcp	3.2.1	3,9 Mo	3,9 Mo	GPL	P
eclipse-source	3.2.1	29 Mo	29 Mo	GPL	P
ant	1.6.5	1 Mo	1 Mo	GPL	P
ant-optional	1.6.5	690 Ko	690 Ko	GPL	P
java-common	0.25	75 Ko	75 Ko	GPL	P
junit	3.8.1.1	70 Ko	70 Ko	GPL	P
libbcel-java	5.1	450 Ko	450 Ko	GPL	B
libcommons-beanutils-java	1.7.0	175 Ko	175 Ko	GPL	B
libcommons-collections3-java	3.1a	475 Ko	475 Ko	GPL	B
libcommons-collections-java	2.1.1	150 Ko	150 Ko	GPL	B
libcommons-dbcp-java	1.2.1	100 Ko	100 Ko	GPL	B
libcommons-digester-java	1.7.1	360 Ko	360 Ko	GPL	B
libcommons-el-java	1.0	220 Ko	220 Ko	GPL	B
libcommons-launcher-java	1.1	130 Ko	130 Ko	GPL	B
libcommons-logging-java	1.0.4	120 Ko	120 Ko	GPL	B
libcommons-modeler-java	1.1	106 Ko	106 Ko	GPL	B
libcommons-pool-java	1.2	41 Ko	41 Ko	GPL	B
libjsch-java	0.1.28	166 Ko	166 Ko	GPL	B
liblog4j1.2-java	1.2.13	216 Ko	216 Ko	GPL	B
liblucene-java	1.4.3	337 Ko	337 Ko	GPL	B
liblucene-java-doc	1.4.3	505 Ko	505 Ko	GPL	B
libmx4j-java	3.0.1	885 Ko	885 Ko	GPL	B
libregexp-java	1.4	87 Ko	87 Ko	GPL	B
libservlet2.4-java	5.0.30	937 Ko	937 Ko	GPL	B
libswt3.2-gtk-java	3.2.1	1,6 Mo	1,6 Mo	GPL	B
libswt3.2-gtk-jni	3.2.1	190 Ko	190 Ko	GPL	B
libtomcat5.5-java	5.5.17	2,3 Mo	2,3 Mo	GPL	B

Note : Concernant la colonne “Type”, le **P** indique un programme et le **B** une bibliothèque, généralement une dépendance.

4.4.6 Profile “Développement Java”

Référence : alc-edi-java

Définition : Environnement de Développement Intégré Java.

Taille sur le DVD : 88,6 Mo

Taille installé : 88,6 Mo

Contenu :

Nom	Version	Taille installé	Taille Paquet	Licence	Type
sun-java5-bin	1.5.0-08	22 Mo	22 Mo	GPL	P
sun-java5-doc*	1.5.0-08	45 Mo	45 Mo	GPL	P
sun-java5-jre	1.5.0-08	7,2 Mo	7,2 Mo	GPL	P
sun-java5-demo	1.5.0-08	9,5 Mo	9,5 Mo	GPL	P
sun-java5-jdk	1.5.0-08	4,9 Mo	4,9 Mo	GPL	P
libltdl3	1.5.22	4,9 Mo	4,9 Mo	GPL	B
java-common	0.25	4,9 Mo	4,9 Mo	GPL	P
odbcinst1debian1	2.2.11	4,9 Mo	4,9 Mo	GPL	P
unixodbc	2.2.11	4,9 Mo	4,9 Mo	GPL	P

Note : Concernant la colonne "Type", le **P** indique un programme et le **B** une bibliothèque, généralement une dépendance.

4.4.7 Profile "Développement Python"

Référence : alc-edi-python

Définition : Environnement de Développement Intégré Python.

Taille sur le DVD : 4 Mo

Taille installé : 4 Mo

Contenu :

Nom	Version	Taille installé	Taille Paquet	Licence	Type
bicylcrepair	0.9-4	109 Ko	109 Ko	GPL	P
python2.4-dev	2.4.4	1,5 Mo	1,5 Mo	GPL	P
python-dev	2.4.3	1 Ko	1 Ko	GPL	P
libcommons-codec-java	1.3-4	52 Ko	52 Ko	GPL	B
eclipse-pydev	1.2.0	2,4 Mo	2,4 Mo	GPL	P

Note : Concernant la colonne "Type", le **P** indique un programme et le **B** une bibliothèque, généralement une dépendance.

4.4.8 Profile "Sécurité Réseau"

Référence : alc-network-tools

Définition : Utilitaires pour l'audit réseau.

Taille sur le DVD : 1,5 Mo

Taille installé : 1,5 Mo

Contenu :

Nom	Version	Taille installé	Taille Paquet	Licence	Type
nmap	4.10-1	700 Ko	700 Ko	GPL	P
ethereal	0.99.3a	21 Ko	21 Ko	GPL	P
ettercap-gtk	1.3a0.7.3-1	226 Ko	226 Ko	GPL	P
ettercap-common	1.3a0.7.3-1	290 Ko	290 Ko	GPL	P

Note : Concernant la colonne "Type", le **P** indique un programme et le **B** une bibliothèque, généralement une dépendance.

4.4.9 Profile “Développement Web”

Référence : alc-web-development

Définition : Outils nécessaires pour le développement Web.

Taille sur le DVD : 39 Mo

Taille installé : 39 Mo

Contenu :

Nom	Version	Taille installé	Taille Paquet	Licence	Type
apache2	2.0.55	36 Ko	36 Ko	GPL	P
apache2-common	2.0.55	788 Ko	788 Ko	GPL	P
apache2-mpm-prefork	2.0.55	201 Ko	201 Ko	GPL	P
apache2-utils	2.0.55	91 Ko	91 Ko	GPL	P
libapache2-mod-php5	5.1.6	2,3 Mo	2,3 Mo	GPL	B
libdbd-mysql-perl	3.0006-1	136 Ko	136 Ko	GPL	B
libdbi-perl	1.51-2	626 Ko	626 Ko	GPL	B
libnet-daemon-perl	0.38-1.1	45 Ko	45 Ko	GPL	B
liblprpc-perl	0.2017-1.1	35 Ko	35 Ko	GPL	B
mysql-client-5.0	5.0.24a-9	6,7 Mo	6,7 Mo	GPL	P
mysql-server-5.0	5.0.24a-9	24 Mo	24 Mo	GPL	P
mysql-server	5.0.24a-9	39 Ko	39 Ko	GPL	P
php5	5.1.6	1 Ko	1 Ko	GPL	P
php5-common	5.1.6	136 ko	136 Ko	GPL	P
php5-mysql	5.1.6	23 Ko	23 Ko	GPL	P
php5-mysqli	5.1.6	39 Ko	39 Ko	GPL	P
phpmyadmin	2.8.2	3,5 Mo	3,5 Mo	GPL	P

Note : Concernant la colonne “Type”, le **P** indique un programme et le **B** une bibliothèque, généralement une dépendance.

4.4.10 Profile “Retouche d’images”

Référence : alc-graphic-tools

Définition : Outils pour la retouche d’images.

Taille sur le DVD : 9 Mo

Taille installé : 9 Mo

Contenu :

Nom	Version	Taille installé	Taille Paquet	Licence	Type
inkscape	0.44	7,5 Mo	7,5 Mo	GPL	P
libglibmm-2.4-1c2a	2.12.2	135 Ko	135 ko	GPL	B
libgtkmm-2.4-1c2a	2.10.2	1,1 Mo	1,1 Mo	GPL	B
libcairomm-1.0	1.2.0	39 Ko	39 Ko	GPL	B

Note : Concernant la colonne “Type”, le **P** indique un programme et le **B** une bibliothèque, généralement une dépendance.

4.4.11 Profile "Édition Vidéo"

Référence : alc-video-edition

Définition : Utilitaire de montage vidéo.

Taille sur le DVD : 1,2 Mo

Taille installé : 1,2 Mo

Contenu :

Nom	Version	Taille installé	Taille Paquet	Licence	Type
kino	0.90	1,1 Mo	1,1 Mo	GPL	P
libsamplerate0	0.1.2	105 Ko	105 Ko	GPL	B

Note : Concernant la colonne "Type", le **P** indique un programme et le **B** une bibliothèque, généralement une dépendance.

4.4.12 Profile "Compatibilité Windows"

Référence : alc-windows-api

Définition : Couche de compatibilité pour lancer des applications win32 nativement.

Taille sur le DVD : 9,2 Mo

Taille installé : 9,2 Mo

Contenu :

Nom	Version	Taille installé	Taille Paquet	Licence	Type
wine	0.9.22	9,2 Mo	9,2 Mo	GPL	P
libartsc0	1.5.4	13 Ko	9,2 Mo	GPL	B

Note : Concernant la colonne "Type", le **P** indique un programme et le **B** une bibliothèque, généralement une dépendance.

4.4.13 Profile "Outils téléchargement"

Référence : alc-download-soft

Définition : Utilitaire de téléchargement P2P.

Taille sur le DVD : 2,7 Mo

Taille installé : 2,7 Mo

Contenu :

Nom	Version	Taille installé	Taille Paquet	Licence	Type
amule	2.1.3	1,3 Mo	1,3 Mo	GPL	P
amule-common	2.1.3	1,5 Mo	1,5 Mo	GPL	P
libcrypto++5.2c2a	5.2.1c2a	1,2 Mo	1,2 Mo	GPL	B
libcrypto++5.2c2a	5.2.1c2a	1,2 Mo	1,2 Mo	GPL	B
libwxbase2.6	2.6.3.2.1.5	512 Ko	512 Ko	GPL	B
libwxgtk2.6	2.6.3.2.1.5	2,6 Mo	2,6 Mo	GPL	B

Note : Concernant la colonne "Type", le **P** indique un programme et le **B** une bibliothèque, généralement une dépendance.

4.4.14 Profile "Lecteur video secondaire"

Référence : alc-videoplayer

Définition : Lecteurs multimédia secondaires.

Taille sur le DVD : 3,5 Mo

Taille installé : 3,5 Mo

Contenu :

Nom	Version	Taille installé	Taille Paquet	Licence	Type
mplayer	0.99+1.0pre8	3,4 Mo	3,4 Mo	GPL	P
mplayer-skins	2-6	39 Ko	39 Ko	GPL	P

Note : Concernant la colonne "Type", le **P** indique un programme et le **B** une bibliothèque, généralement une dépendance.

4.4.15 Profile "SoftPhones"

Référence : alc-softphones

Définition : Utilitaires de Téléphonie sur IP (VOIP).

Taille sur le DVD : 17 Mo

Taille installé : 17 Mo

Contenu :

Nom	Version	Taille installé	Taille Paquet	Licence	Type
librtel	0.5.1-0.1	333 Ko	333 Ko	GPL	B
skype*	1.3.0.53	9,6 Mo	9,6 Mo	GPL	P
liba52-0.7.4	0.7.4-3	26 Ko	26 Ko	GPL	B
libavcodec0d	3a0 (cvs20060823)	1,5 Mo	1,5 Mo	GPL	B
libdc1394-13	1.1.0	31 Ko	31 Ko	GPL	B
libgsm1	1.0.10-13	28 Ko	28 Ko	GPL	B
wengophone	0.99 (svn4511)	2,3 Mo	2,3 Mo	GPL	P
libqt3-mt	3a3.3.6	3 Mo	3 Mo	GPL	B

Note : Concernant la colonne "Type", le **P** indique un programme et le **B** une bibliothèque, généralement une dépendance.

4.4.16 Profile "Pack fontes"

Référence : alc-fonts

Définition : Ajout de nouvelles polices (dont Microsoft).

Taille sur le DVD : 23 Ko

Taille installé : 23 Ko

Contenu :

Nom	Version	Taille installé	Taille Paquet	Licence	Type
msttcorefonts*	1.2	23 Ko	23 Ko	GPL	P

Note : Concernant la colonne "Type", le **P** indique un programme et le **B** une bibliothèque, généralement une dépendance.

5 Spécifications d’interface

5.1 Découpage de l’application en modules

Il est nécessaire de découper l’application selon les tâches dites importantes, voici le découpage que nous avons choisi de faire :

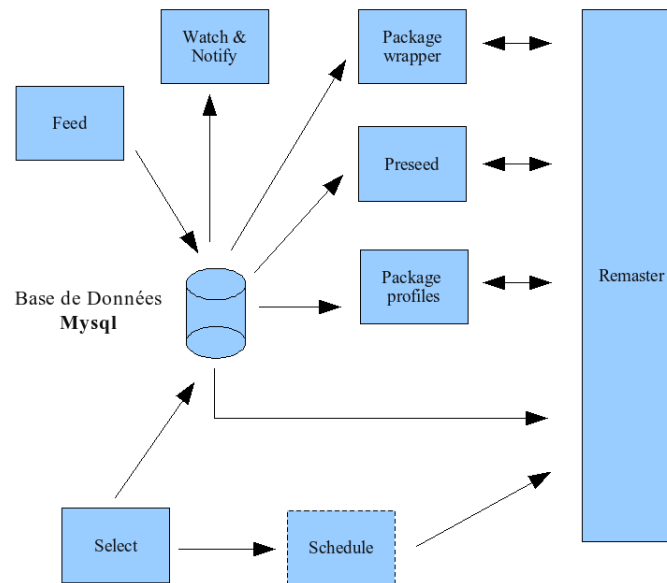


FIG. 8 – Schéma – Découpage de l’application OS A La Carte en modules.

Explication sur le découpage en modules :

- **“Feed”** prendra en compte le suivi des mises à jour proposées par les dépôts officiels d’Ubuntu (Réponse à la contrainte de *sécurité*). Ce module sera en relation avec la base de données et donc défini dans la partie “*Interfaces Logiciel – Logiciel*” ;
- **“Watch and Notify”** est le module qui est lié au module “Feed”, il ne concerne pas directement la personnalisation du DVD d’installation. Ce module sera en relation avec la base de données et donc défini dans la partie “*Interfaces Logiciel – Logiciel*” ;
- **“Package Profiles”** s’occupera de la gestion des profils d’applications. Ce module sera en relation avec la base de données et donc défini dans la partie “*Interfaces Logiciel – Logiciel*” ;
- **“Package Wrapper”** gèrera les paquets, à savoir leur ouverture, leur modification, et finalement leur reconstruction. Ce module sera en relation avec la base de données et donc défini dans la partie “*Interfaces Logiciel – Logiciel*” ;
- **“Preseed”** concernera le paramétrage du DVD d’installation. Ce module sera en relation avec la base de données et donc défini dans la partie “*Interfaces Logiciel – Logiciel*” ;

- “**Remaster**” aura pour tâche la création de l’image ISO en prenant en compte les modules “Package Profile”, “Package Wrapper” et “Preseed”. Ce module sera en relation avec la base de données et donc défini dans la partie “*Interfaces Logiciel – Logiciel*” ;
- “**Select**” sera l’interface ou l’utilisateur sélectionnera ses choix. Ce module sera en relation avec l’utilisateur et donc défini dans la partie “*Interfaces Homme – Machine*” ;
- “**Schedule**” gèrera les utilisateurs, avec une file d’attente, la sauvegarde des sélections des utilisateurs (Module Non Prioritaire). Ce module sera en relation avec la base de données et donc défini dans la partie “*Interfaces Logiciel – Logiciel*”.

On trouvera deux types de modules, les modules qui auront un accès en **lecture seule** et ceux qui auront un accès en **lecture/écriture** à la base de donnée.

Nom du module	Type d’accès à la Base de Données
“Feed”	Lecture/Écriture
“Watch & Notify”	Lecture seule
“Package wrapper”	Lecture seule
“Preseed”	Lecture seule
“Package profiles”	Lecture seule
“Remaster”	Lecture seule
“Select”	Lecture/Écriture

FIG. 9 – Tableau – Récapitulatif des modules et de leur type d’accès à la base de données.

5.2 Interfaces Homme – Machine

Cette spécification d'interface concerne le module "Select".

L'objectif est de configurer tout depuis une seule page internet.

Pour que l'utilisateur ne se **perde** pas dans tous les paramètres, nous allons afficher une page web **simple**. On retrouve la division en 3 parties : la configuration du système de base (au niveau du Preseed), la configuration détaillée de paquets et enfin la sélection de profils d'applications.

The diagram illustrates a web interface for user configuration, enclosed in a rectangular frame. At the top is a light blue header bar labeled "Message d'accueil". Below this, the interface is divided into three main sections, each with a title and a light blue background:

- Configuration du système de Base (au niveau du Preseed)**: Contains three input fields labeled "Nom utilisateur", "Mot de passe", and "Choix langue".
- Configuration détaillée des paquets**: Contains a list of items, each preceded by a right-pointing triangle (▶):
 - Personnalisation du navigateur internet
 - Personnalisation de l'éditeur de texte
 - Personnalisation du terminal
 - Personnalisation du bureau
 - Personnalisation des bureaux virtuels
 - Personnalisation de l'interpréteur de commandes
- Sélection des profils d'applications**: Contains a list of items, each preceded by a checkmark (✓):
 - Extraction des archives de différents formats.
 - Lecture de vidéos et sons en différents formats
 - Environnement de Développement Intégré Java
 - Environnement de Développement Intégré Python
 - Utilitaire de montage vidéo
 - ...

At the bottom of the frame is a light blue button labeled "Valider la Sélection".

Les cases à cocher sont représentées par des ✓.

FIG. 10 – IHM – Interface non détaillée de sélection de configuration de l'utilisateur .

Cependant, lorsque l'utilisateur souhaitera **plus de détails** concernant la configuration des paquets notamment, il cliquera sur la rubrique ce qui aura pour rôle d'**afficher** une partie détaillée. Cette partie est représentée dans l'IHM suivante en verte :

Message d'accueil

Configuration du système de Base (au niveau du Preseed)

Nom utilisateur

Mot de passe

Choix langue

Configuration détaillée des paquets

▼ Personnalisation du navigateur internet

☒ Utiliser l'ancien mode de fermeture des onglets .
☒ Cacher le bouton "Go".
☒ Forcer l'ouverture des liens "externes" dans un nouvel onglet.
☒ Désactiver la demande de fermeture des onglets.
☒ Activer le bouton du milieu.
☒ Activer le défilement avec le bouton du milieu.
☒ Désactiver le préchargement des sites.
☒ Désactiver le cache du disque.
☒ Désactiver l'historique de navigation.
☒ Désactiver les cookies.

► Personnalisation de l'éditeur de texte
 ► Personnalisation du terminal
 ► Personnalisation du bureau
 ► Personnalisation des bureaux virtuels
 ► Personnalisation de l'interpréteur de commandes

Sélection des profils d'applications

☒ Extraction des archives de différents formats.
☒ Lecture de vidéos et sons en différents formats
☒ Environnement de Développement Intégré Java
☒ Environnement de Développement Intégré Python
☒ Utilitaire de montage vidéo
 ...

Valider la Sélection

Les cases à cocher sont représentées par des √.

FIG. 11 – IHM – Interface détaillée de sélection de configuration de l'utilisateur .

Ainsi, les informations de configuration des paquets sont toutes **masquées** par défaut et on peut donc dérouler toutes les options de configuration des paquets.

5.3 Interfaces Logiciel – Logiciel

Cette spécification d'interface concerne la plupart des modules. Les modules “Watch & Notify”, “Package wrapper”, “Preseed”, “Package profiles” et “Remaster” seront en relation avec les autres par l'intermédiaire de la base de données.

Pour que l'information soit **cohérente** entre les différents modules nous utiliserons un **identifiant**. Cet identifiant sera créé dès que le client commencera à utiliser notre programme et donc dans le module

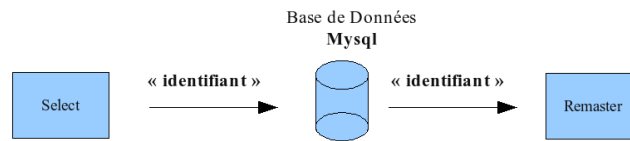


FIG. 12 – Schéma – Communication entre les modules "Select" et "Feed" à l'aide d'un identifiant.

"Select".

Ainsi chaque module commencera par vérifier que l'identifiant fourni est valide dans la base de données, puis que les informations qui lui sont associées le sont également.

ISAIP
18, rue du 8 Mai 1945
49000 St Barthélémy d'Anjou



Dossier de Conception

Développement d'une application de personnalisation de système
d'exploitation : "OS A La Carte"

1 Février 2007

Mohammed BADISS, Jérôme CHARLOT
2^{ème} année CPI
Promotion 2005 – 2007

Sous la responsabilité de :
M Alexandre LEPRIEULT ;



Table des matières

1	Traçabilité	4
2	Introduction	4
2.1	But du document	4
2.2	Documentation	4
2.2.1	Documents de référence	4
3	Rappels	4
3.1	Objectifs du projet	4
3.2	Procédé de fabrication	5
3.3	Les Fonctionnalités	6
3.4	Découpage de l’application en modules	7
3.5	Rôle de chaque module	7
4	Architecture dynamique du logiciel	9
4.1	Modèle Physique de Données	9
4.2	Tâche de préparation	10
4.2.1	Traitements effectués	10
4.2.2	Enchaînements et flots de données	10
4.2.3	Modules utilisés	10
4.3	Tâche de mise à jour des paquets	10
4.3.1	Traitements effectués	10
4.3.2	Enchaînements et flots de données	10
4.3.3	Modules utilisés	11
5	Conception des modules	12
5.1	Module “Feed”	12
5.1.1	Objectifs du module	12
5.1.2	Relations d’utilisation avec d’autres modules	12
5.2	Conception du module “Feed”	13
5.2.1	Procédure de vérification des informations fournies en paramètre.	13
5.2.2	Traitement des dépôts et alimentation de la base de données.	14
5.2.3	Définitions de types	15
5.2.4	Dépendances du modules	16
5.3	Module “Watch and Notify”	17
5.3.1	Objectifs du module	17
5.3.2	Relations d’utilisation avec d’autres modules	17
5.3.3	Définitions de types	17
5.4	Module “Package Profiles”	18
5.4.1	Objectifs du module	18
5.4.2	Relations d’utilisation avec d’autres modules	18
5.5	Conception du module “Package Profile”	19
5.5.1	Définitions de types	19
5.5.2	Dépendances du modules	19
5.6	Module “Package Wrapper”	20
5.6.1	Objectifs du module	20
5.6.2	Relations d’utilisation avec d’autres modules	20

5.6.3	Définitions de types	20
5.6.4	Dépendances du modules	20
5.7	Module “Preseed”	21
5.7.1	Objectifs du module	21
5.7.2	Relations d’utilisation avec d’autres modules	21
5.8	Conception du module “Preseed”	22
5.8.1	Définitions de types	22
5.8.2	Dépendances du modules	22
5.9	Module “Remaster”	23
5.9.1	Objectifs du module	23
5.9.2	Relations d’utilisation avec d’autres modules	23
5.10	Conception du script “Préparation Remaster”	23
5.11	Conception du module “Remaster”	24
5.11.1	Définitions de types	24
5.11.2	Dépendances du modules	25
5.12	Module “Select”	26
5.12.1	Objectifs du module	26
5.12.2	Relations d’utilisation avec d’autres modules	27
5.13	Conception de l’interface Homme – Machine	27
5.14	Procédure de vérification de la sélection de l’utilisateur	28
5.15	Utilisation de fonctions PHP	28
5.15.1	Définitions de types	29

Table des figures

1	Schéma – Déroulement du procédé de fabrication.	5
2	Schéma – Choix possibles de l'utilisateur.	6
3	Schéma – Découpage de l'application OS A La Carte en modules.	7
4	Schéma – Modèle Physique de Données pour l'application OS A La Carte.	9
5	Schéma – Conception du module “Feed”.	13
6	Schéma – Exemple d'un fichier dépôt contenant une liste de paquets.	15
7	Tableau – Liste des variables et leur type du module “Feed”	16
8	Tableau – Liste des variables et leur type du module “Watch and Notify”	18
9	Schéma – Conception du module “Package Profile”.	19
10	Schéma – Conception du module “Preseed”.	22
11	Schéma – Conception du script “Préparation Remaster”.	23
12	Schéma – Conception du module “Remaster”.	24
13	IHM – Interface non détaillée de sélection de configuration de l'utilisateur	26
14	IHM – Interface détaillée de sélection de configuration de l'utilisateur	27

1 Traçabilité

Version	Date	Commentaires
1.00	05/01/07	Première version
1.20	21/01/07	Ajout des schémas de conception
1.30	03/02/07	Ajout de la partie Rappel
1.40	03/02/07	Redéfinition des modules “Select” et “Feed”
2.00	04/02/07	Correction orthographe

2 Introduction

2.1 But du document

Ce dossier présente la phase de conception du logiciel d’un point de vue méthodologique. Il dépend très peu de l’application elle-même.

2.2 Documentation

2.2.1 Documents de référence

Les documents de référence sont le dossier de **Cahier des Charges** et la dossier de **Specifications Fonctionnelles**.

3 Rappels

Il est nécessaire d’effectuer quelques rappels, pour rappeler les parties importantes du déroulement du projet.

3.1 Objectifs du projet

Notre projet nommé “OS A La Carte” a pour but de **personnaliser** un système d’exploitation depuis une page Internet, plus particulièrement de générer une image du DVD d’installation d’une distribution Linux.

Cette image au format ISO est destinée a être gravée sur un DVD. Une fois gravée, l’utilisateur insère le DVD dans son lecteur de DVD puis amorce le processus d’installation. L’utilisateur installe alors ce système d’exploitation sur sa machine.

Notre travail repose donc entièrement sur la **modification** selon les demandes et les besoins du client de la distribution Linux nommée **Ubuntu**.

Cette **personnalisation** s’effectue à la base du système d’exploitation, tout d’abord au niveau des programmes qui sont couramment utilisés et mais également au niveau de l’installation d’applications additionnelles.

3.2 Procédé de fabrication

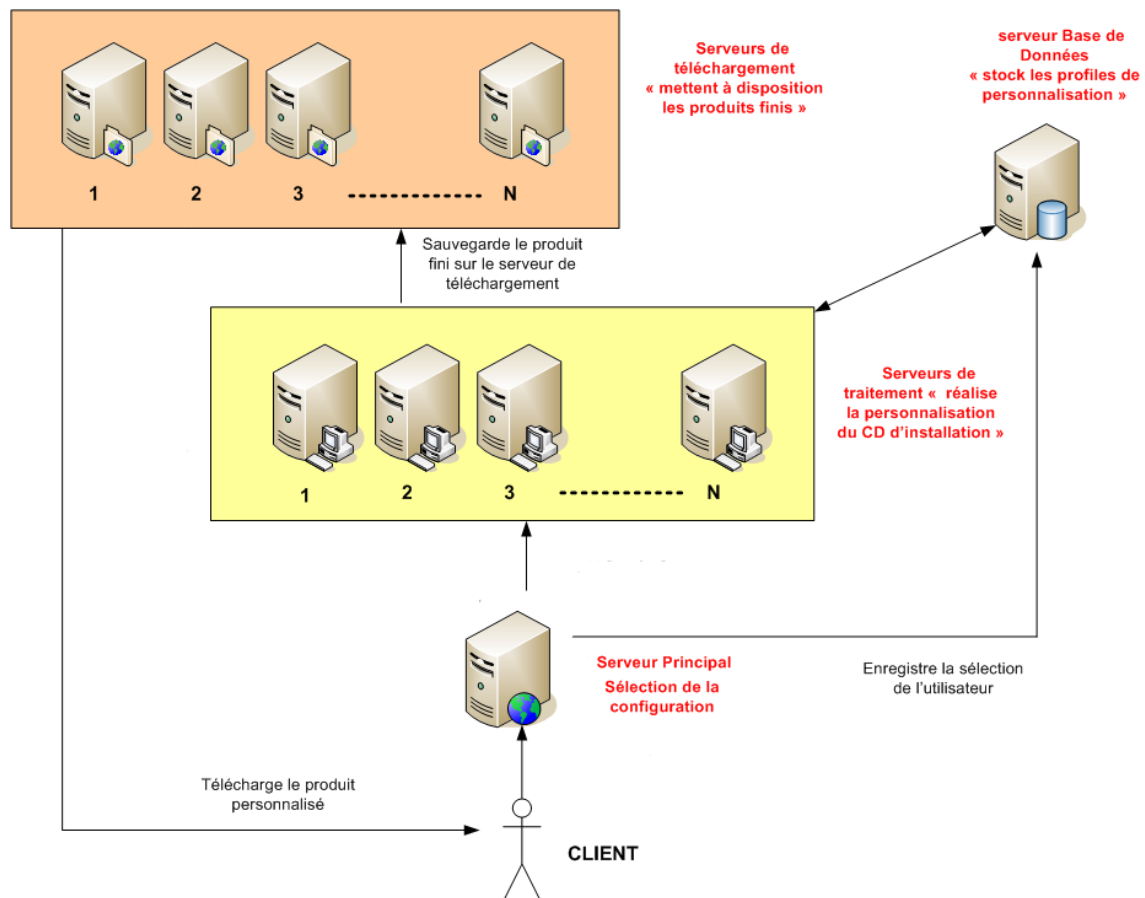


FIG. 1 – Schéma – Déroulement du procédé de fabrication.

Le procédé de fabrication se déroule en **trois** phases, qui sont les suivantes :

1. La première phase est la **sélection** de la configuration depuis notre interface Web. Cette opération est la seule effectuée par l'utilisateur. Le temps de réponse l'interface web dépend principalement de la bande passante du client, et de notre serveur. Mais avec les connexions actuelles, la technologie ADSL étant largement répandue, ce temps est **négligeable** (à peine quelques secondes).
2. Une fois la phase de personnalisation terminée. La phase de **génération** du DVD d'installation, réalisée par notre application, peut exiger environ **3** minutes sur le serveur dédié (un Pentium 4 ou équivalent).
3. Finalement, la phase de **téléchargement** de l'image du DVD est la plus **exigeante** au niveau temps, car comme la première phase, elle dépend de la bande passante que dispose le client mais surtout celle que nous allouons pour le serveur de téléchargement.

3.3 Les Fonctionnalités

Voici un schéma qui représente l'étendue des choix pour lesquels l'utilisateur peut personnaliser son système d'exploitation :

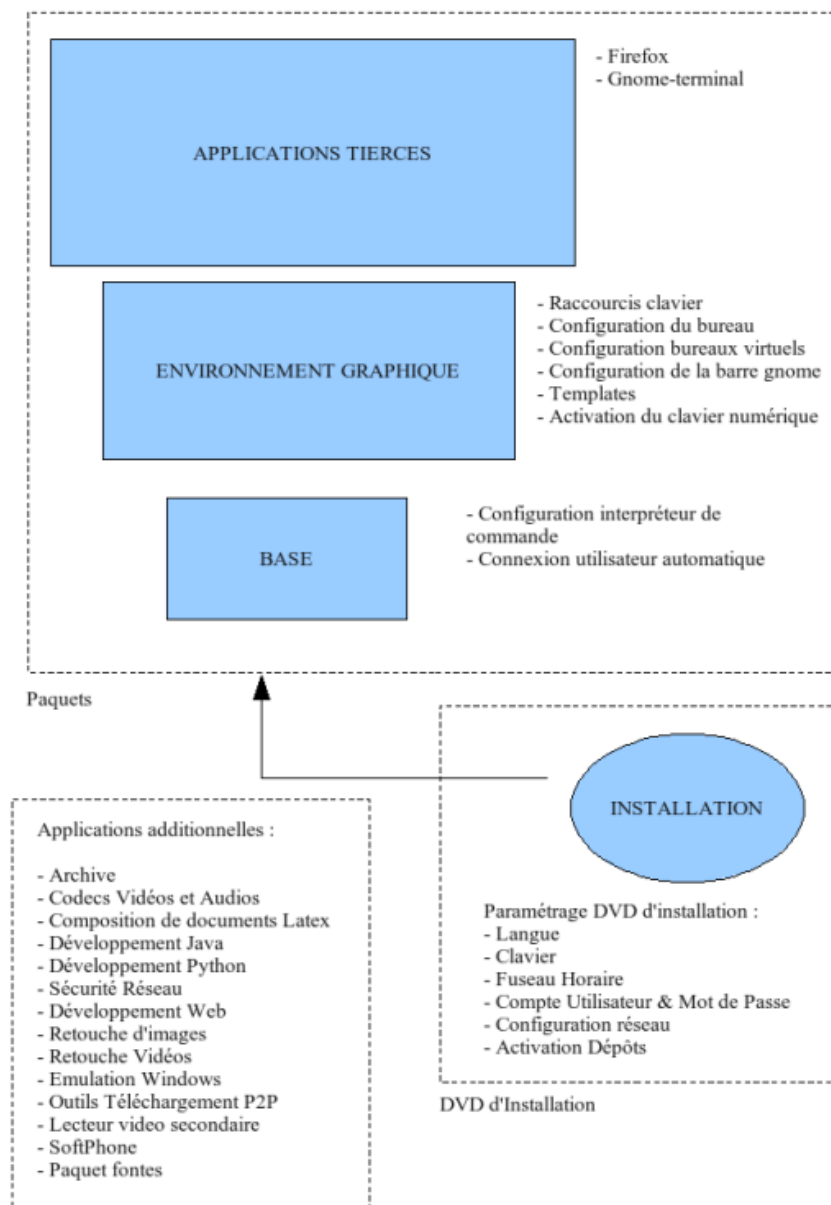


FIG. 2 – Schéma – Choix possibles de l'utilisateur.

L'utilisateur personnalise le DVD d'installation à **trois** niveaux différents.

Les trois niveaux sont :

1. Fonctionnalités au niveau du DVD d'installation.
2. Modifications des paquets les plus importants.
3. Choix des profils d'applications.

Ces différents niveaux sont **abstraits** et **transparents** pour l'utilisateur, ils correspondent à un découpage en modules de l'application. Ce découpage est détaillé dans la partie “*Découpage de l'application en modules*”.

3.4 Découpage de l'application en modules

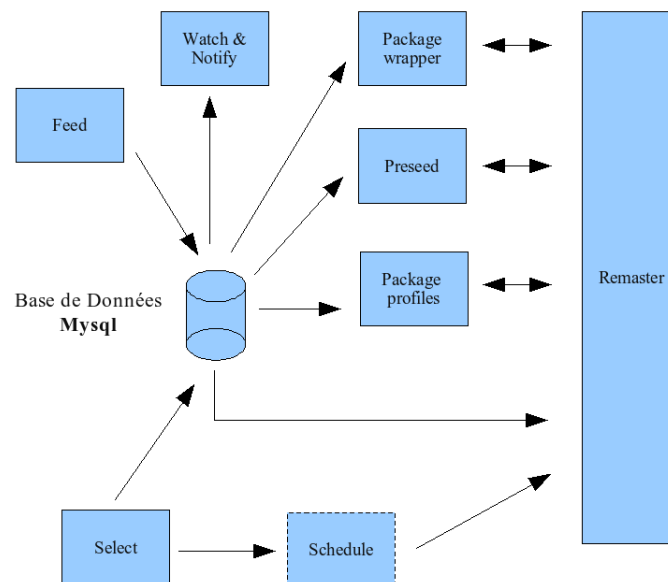


FIG. 3 – Schéma – Découpage de l'application OS A La Carte en modules.

3.5 Rôle de chaque module

- “**Feed**” prendra en compte le suivi des mises à jour proposées par les dépôts officiels d'Ubuntu (Réponse à la contrainte de *sécurité*) ;
- “**Watch and Notify**” aura pour rôle d'afficher les changements des dépôts ;
- “**Package Profiles**” s'occupera de la gestion des profils d'applications ;
- “**Package Wrapper**” gèrera les paquets, à savoir leur ouverture, leur modification, et finalement leur reconstruction ;
- “**Preseed**” concernera le paramétrage du CD d'installation ;
- “**Remaster**” aura pour tâche la création de l'image ISO en prenant en compte les modules “**Package Profile**”, “**Package Wrapper**” et “**Preseed**” ;

- **“Select”** sera l’interface ou l’utilisateur sélectionnera ses choix ;
- **“Schedule”** gèrera les utilisateurs, avec une file d’attente, la sauvegarde des sélections des utilisateurs (Module Non Prioritaire).

4 Architecture dynamique du logiciel

4.1 Modèle Physique de Données

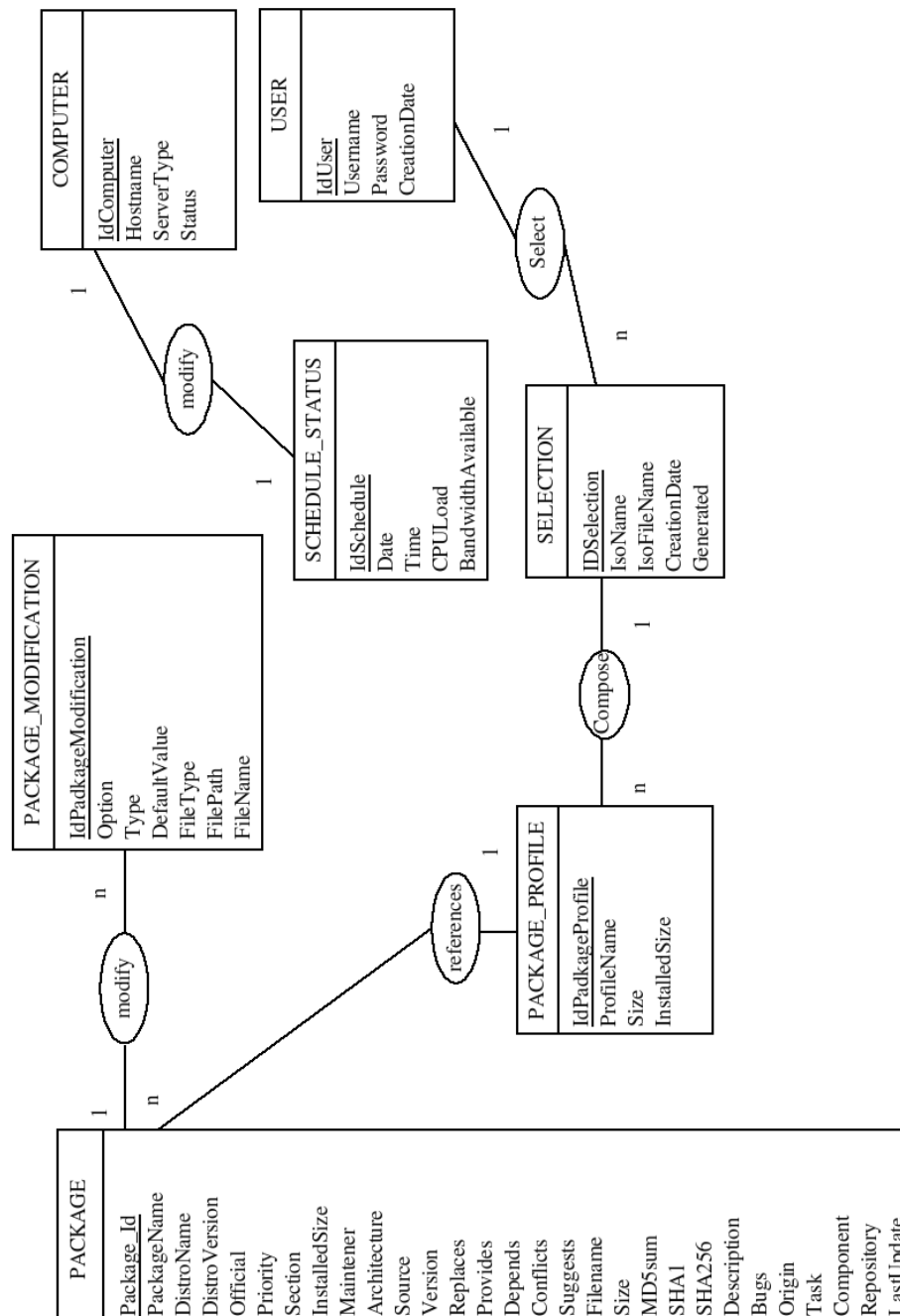


FIG. 4 – Schéma – Modèle Physique de Données pour l'application OS A La Carte.

Le **Modèle Physique de Données** est un modèle qui tient compte du moteur de base données pour gérer les données. Ainsi avec le MPD nous savons quels sont les types de chaque attribut et également les clés étrangères.

4.2 Tâche de préparation

4.2.1 Traitements effectués

Cette tâche est effectuée par l'administrateur, elle est **exceptionnelle** car réalisée qu'une seule fois lors de l'installation de l'application.

Elle consiste à **préparer** les répertoires et autres fichiers qui seront utilisés par la suite par le programme de génération de l'image DVD.

C'est également durant cette étape que nous allons utiliser la **cryptographie** avec l'aide des clés publique et privée. En effet lorsque le processus d'installation est en cours d'exécution, il va vérifier que les programmes présents sur le DVD d'installation n'ont pas été modifiés pour installer des applications dites “compromises” (contenant des trojans, backdoor, bombes logiques, ...).

Ainsi il est nécessaire de signer l'image DVD et de joindre la clé publique dans l'image. Cette étape est réalisée pendant la **tâche de préparation**.

4.2.2 Enchaînements et flots de données

Une fois tout le traitement réalisé, l'administrateur regroupe les fichiers sous forme d'archives. Ce seront ces archives qui seront utilisées comme **références** pour construire l'image du DVD d'installation.

4.2.3 Modules utilisés

Les modules qui seront utilisés sont “Feed” et “Remaster”.

4.3 Tâche de mise à jour des paquets

4.3.1 Traitements effectués

Notre image du DVD se base sur Ubuntu “Edgy Eft” qui est sortie au mois d'octobre 2006, or depuis ce temps il y a eu des **misés à jour** qui ont été diffusées sur les dépôts.

Tous les laps de temps réguliers, par exemple toutes les 6 heures ou 12 heures, nous réactualiserons le contenu de dépôts “Sécurité” et mettrons à jour notre archive de référence.

4.3.2 Enchaînements et flots de données

Le module “Feed” réactualisera les nouveaux paquets disponibles et une requête sur la base de données fera ressortir les paquets qui seront présent sur l'image DVD.

Il nous restera donc à mettre à jour les paquets sélectionnés et la liste qui les référence tous.

4.3.3 Modules utilisés

Les modules qui seront en relation avec cette tâche seront “Remaster”, “Watch and Notify” et “Feed”

5 Conception des modules

5.1 Module “Feed”

5.1.1 Objectifs du module

Ce module prendra en compte le suivi des mises à jour proposées par les dépôts officiels d’Ubuntu (Réponse à la contrainte de *sécurité*).

Ce module sera développé en langage procédurale **Python**.

5.1.2 Relations d’utilisation avec d’autres modules

Ce module bien qu’étant très fortement lié au module “**Watch & Notify**” n’est pas en relation avec d’autres modules.

Cependant ce module sera en relation avec la base de données.

5.2 Conception du module “Feed”

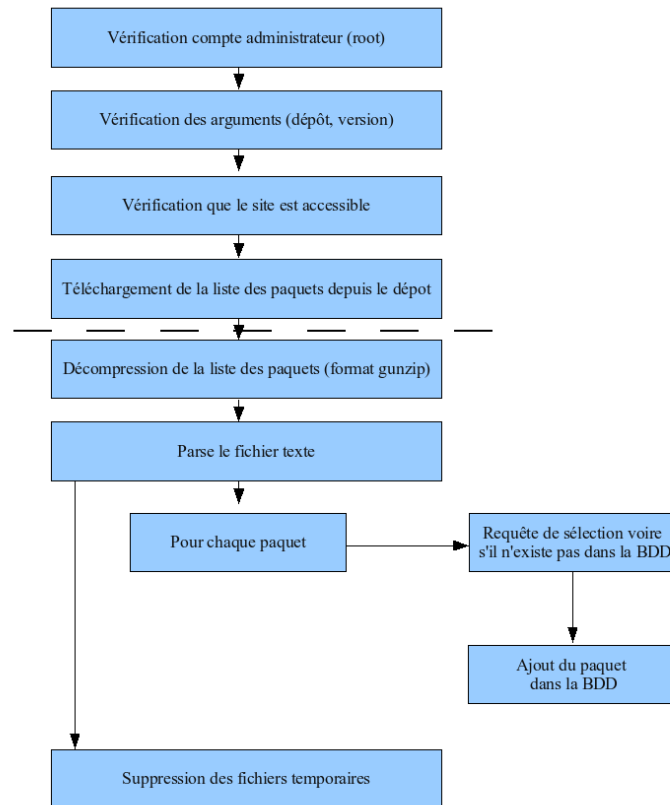


FIG. 5 – Schéma – Conception du module “Feed”.

Le modules “Feed” fonctionne indépendamment des autres modules et va rechercher des informations dans la base de données et en ajouter et/ou en mettre à jour.

5.2.1 Procédure de vérification des informations fournies en paramètre.

Tout d’abord le programme “Feed” pour fonctionner a besoin de **3** paramètres qui sont les suivants :

- l’**architecture** (par exemple : i386, ppc ou amd64) ;
- le nom du **dépôt** (par exemple : edgy, edgy-updates, edgy-security ou edgy-backports) ;
- les **composants** de chaque dépôts (par exemple : main, restricted, multiverse, universe).

Pour effectuer la vérification, tout d’abord nous comptons le nombre d’arguments avec le module **sys.argv** il faut que ce tableau ait une longueur de **4** éléments (le nom du programme et les 3 arguments).

Ensuite, nous essayons de vérifier que les informations fournies sont exactes, pour cela nous allons vérifier que le **fichier** correspondant au **composant du dépôt** existe.

Nous allons pour cela, utiliser le module Python **httplib** et les fonctions suivantes :

- **httplib.HTTPConnection()** : qui instancie un objet de type **connexion HTTP** ;
- **request()** : qui émet une **requête** de type "GET" avec les paramètres fournis ;
- **getresponse()** : qui permet de savoir l'**état** de la requête **HTTP** * ;
- **close()** : qui permet de fermer la connexion **HTTP**.

* Note : Le résultat est celui défini par la norme **HTTP**, le code de retour normal est 200, tous les autres codes sont des erreurs (par exemple **404** pour fichier non trouvé).

Ainsi si la fonction **getresponse()** retourne un code différent de **200**, alors c'est que l'utilisateur se sera trompé à saisir les paramètres, ce qui aura pour conséquence d'interrompre le programme avec une sortie d'erreur à l'aide de la fonction **sys.exit("erreur saisie de paramètre")**.

Une fois que nous nous sommes assurés que les paramètres transmis sont valides, nous pouvons passer au traitement des paquets contenu dans les dépôts.

5.2.2 Traitement des dépôts et alimentation de la base de données.

Nous allons à nouveau utiliser le module Python **httplib** pour télécharger localement le fichier qui liste l'ensemble des paquets pour un composant d'un dépôt donné.

Pour cela on utilise les fonctions suivantes :

- **httplib.HTTPConnection()** : qui instancie un objet de type **connexion HTTP** ;
- **request()** : qui émet une **requête** de type "GET" avec les paramètres fournis ;
- **getresponse()** : qui permet de savoir l'**état** de la requête **HTTP** * ;
- **urllib2.Request()** : qui permet de récupérer une copie local d'un fichier accessible sur un serveur web ;
- **request.add_header()** : permet de spécifier le **type de fichier** que nous téléchargerons (le MIME TYPE), il s'agit ici d'une **archive compressée** au format **GZIP** (nous mettrons comme paramètres : 'Accept-encoding', 'gzip') ;
- **close()** : qui permet de fermer la connexion **HTTP**.

Ainsi, à ce stade, nous avons récupéré une copie du fichier qui contient la liste des paquets qui est compressé au format **GZIP**.

Pour traiter cette archive, nous allons utiliser le module Python **gzip**, et les fonctions suivantes :

- Déclaration du fichier en fichier **compressé StringIO.StringIO(compresseddata)** ;
- **gzip.GzipFile(fileobj=compresseddata)**, permet de gérer les fichiers compressés au format **gzip** ;
- **open()** permet d'ouvrir le fichier compressé ;
- **read()** permet de lire le contenu du fichier.

Ensuite, nous lisons le fichier de contenu du dépôt (la liste de ses paquets), on le "parse", on analyse chaque ligne, un paquet est défini par le mot clé **Package** et toutes les lignes qui suivent sont des **informations relatives** à ce paquet.

```
Package: apache2
Priority: optional
Section: web
Installed-Size: 80
Maintainer: Ubuntu Core Developers <ubuntu-devel@lists.ubuntu.com>
Original-Maintainer: Debian Apache Maintainers <debian-apache@lists.debian.org>
Architecture: i386
Version: 2.0.55-4ubuntu4
Depends: apache2-mpm-worker (= 2.0.55-4ubuntu4) | apache2-mpm-prefork (= 2.0.55-
| apache2-mpm-perchild (= 2.0.55-4ubuntu4)
Filename: pool/main/a/apache2/apache2_2.0.55-4ubuntu4_i386.deb
Size: 35976
MD5sum: b41e510af15a616746930c7ba86362c2
SHA1: be0399c61f2b3e3a27a0b1f2f558bc7218b7f7bc
SHA256: 2d1659ecb4512b257c933a2c9842f8a416641f939000f7d105ef9fa30c167a3b
Description: next generation, scalable, extendable web server
 Apache v2 is the next generation of the omnipresent Apache web server. This
 version - a total rewrite - introduces many new improvements, such as
 threading, a new API, IPv6 support, request/response filtering, and more.
 Bugs: mailto:ubuntu-users@lists.ubuntu.com
 Origin: Ubuntu
 Task: lamp-server

Package: apache2-mpm-prefork
Priority: optional
Section: net
Installed-Size: 468
...
```

FIG. 6 – Schéma – Exemple d’un fichier dépôt contenant une liste de paquets.

Avant de créer un nouvel enregistrement dans la base de données, on vérifie que cet enregistrement n’existe pas déjà avec les fonctions du module **Mysqldb** :

- **db.cursor()** : définit un curseur pour manipuler les données de la base de données ;
- **cursor.execute()** : permet d’exécuter une requête au niveau de la base de données et notamment la table “Package” ;
- **cursor.fetchall()** : permet de récupérer le résultat de la requête sous forme d’un tableau en Python.

Une fois que le script Python “Feed” a terminé son exécution, on se retrouve avec une **liste de paquets** dans la base de données qui **correspond** au contenu des **dépôts** d’Ubuntu disponibles sur Internet.

5.2.3 Définitions de types

Ce module est composé des types suivants :

Nom variable	Type de données
Package_id	int(5)
PackageName	varchar(50)
DistroName	varchar(20)
DistroVersion	varchar(6)
Official	tinyint(1)
Priority	varchar(10)
Section	varchar(20)
InstalledSize	int(15)
Maintener	varchar(50)
Architecture	varchar(9)
Source	varchar(45)
Version	varchar(45)
Replaces	varchar(50)
Provides	varchar(75)
Depends	text
Conflicts	varchar(75)
Suggests	text
Filename	varchar(250)
Size	int(15)
MD5sum	varchar(32)
SHA1	varchar(60)
SHA256	varchar(60)
Description	text
Bugs	varchar(75)
Origin	varchar(20)
Task	varchar(125)
Component	varchar(20)
Repository	varchar(20)
LastUpdate	datetime
CD	tinyint(1)
PackageProfileSelection	tinyint(1)

FIG. 7 – Tableau – Liste des variables et leur type du module “Feed”

5.2.4 Dépendances du modules

Pour fonctionner ce module a besoin des dépendances suivantes :

- urllib2
- httplib
- StringIO
- gzip
- sysgzip
- MySQLdb
- time

5.3 Module “Watch and Notify”

5.3.1 Objectifs du module

Ce module aura pour rôle d’afficher le résultat du module “Feed” sous forme d’une page web. Ainsi il ne concerne pas directement la personnalisation du DVD d’installation.

5.3.2 Relations d’utilisation avec d’autres modules

Ce module sera en relation avec la base de données.

5.3.3 Définitions de types

Ainsi ce module partagera les mêmes variables que le module “Feed” qui sont les suivantes :

Nom variable	Type de données
Package_id	int(5)
PackageName	varchar(50)
DistroName	varchar(20)
DistroVersion	varchar(6)
Official	tinyint(1)
Priority	varchar(10)
Section	varchar(20)
InstalledSize	int(15)
Maintener	varchar(50)
Architecture	varchar(9)
Source	varchar(45)
Version	varchar(45)
Replaces	varchar(50)
Provides	varchar(75)
Depends	text
Conflicts	varchar(75)
Suggests	text
Filename	varchar(250)
Size	int(15)
MD5sum	varchar(32)
SHA1	varchar(60)
SHA256	varchar(60)
Description	text
Bugs	varchar(75)
Origin	varchar(20)
Task	varchar(125)
Component	varchar(20)
Repository	varchar(20)
LastUpdate	datetime
CD	tinyint(1)
PackageProfileSelection	tinyint(1)

FIG. 8 – Tableau – Liste des variables et leur type du module “Watch and Notify”

5.4 Module “Package Profiles”

5.4.1 Objectifs du module

Ce module s’occupera de la gestion des profils d’applications. Il aura pour rôle de copier les paquets additionnels sélectionnés par l’utilisateur.

5.4.2 Relations d’utilisation avec d’autres modules

Ce module sera en relation avec la base de données.

5.5 Conception du module “Package Profile”

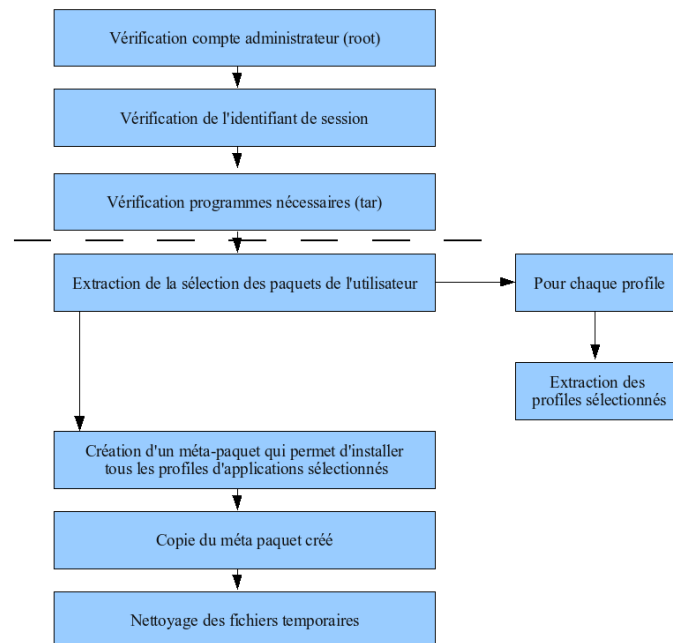


FIG. 9 – Schéma – Conception du module “Package Profile”.

5.5.1 Définitions de types

Ce paragraphe donne la liste des définitions de type. Dans le cas d’une conception orientée objets cette définition de type peut être unique et représenter les attributs de l’objet associé au module.

Nom variable	Type de données
IdPackageModification	int(8)
Option	varchar(250)
Type	tinyint
DefaultValue	varchar(30)
FileType	varchar(10)
FilePath	varchar(100)
FileName	varchar(35)

5.5.2 Dépendances du modules

Pour fonctionner ce module a besoin des dépendances suivantes :

- MySQLdb

5.6 Module “Package Wrapper”

5.6.1 Objectifs du module

Ce module gérera les paquets, à savoir leur ouverture, leur modification, et finalement leur reconstruction.

5.6.2 Relations d’utilisation avec d’autres modules

Ce module sera en relation avec la base de données.

5.6.3 Définitions de types

Ce paragraphe donne la liste des définitions de type. Dans le cas d’une conception orientée objets cette définition de type peut être unique et représenter les attributs de l’objet associé au module.

5.6.4 Dépendances du modules

Pour fonctionner ce module a besoin des dépendances suivantes :

- MySQLdb

5.7 Module “Preseed”

5.7.1 Objectifs du module

Ce module concerne le paramétrage du DVD d’installation avec la réalisation du fichier de configuration.

5.7.2 Relations d’utilisation avec d’autres modules

Ce module sera en relation avec la base de données.

5.8 Conception du module "Preseed"

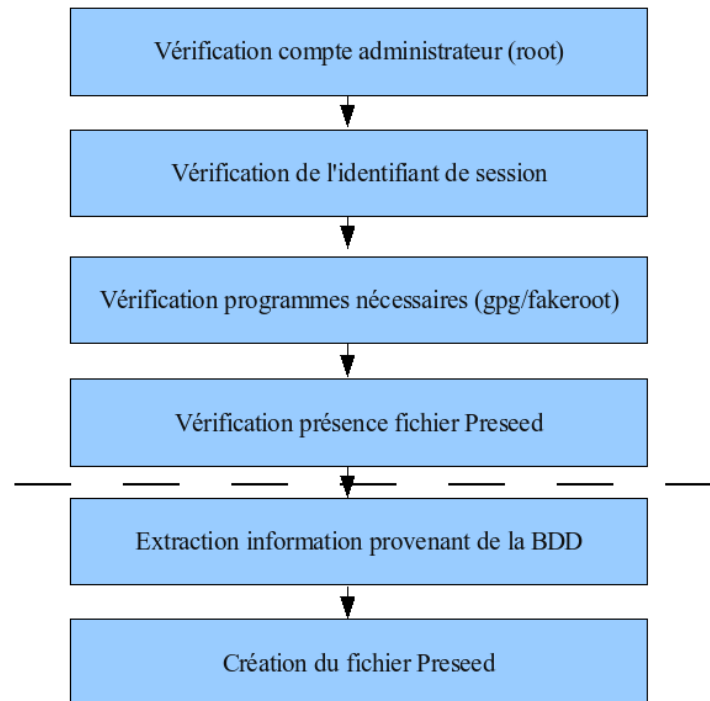


FIG. 10 – Schéma – Conception du module "Preseed".

5.8.1 Définitions de types

Ce paragraphe donne la liste des définitions de type. Dans le cas d'une conception orientée objets cette définition de type peut être unique et représenter les attributs de l'objet associé au module.

Nom variable	Type de données
preseed_lang	varchar(20)
preseed_network	tinyint(1)
preseed_hostname	varchar(20)
preseed_domainname	varchar(20)
preseed_timezone	varchar(45)
preseed_user	varchar(30)
preseed_password	varchar(30)

5.8.2 Dépendances du modules

Pour fonctionner ce module a besoin des dépendances suivantes :

- MySQLdb

5.9 Module “Remaster”

5.9.1 Objectifs du module

Ce module aura pour tâche la création de l’image ISO en prenant en compte les modules “Package Profile”, “Package Wrapper” et “Preseed”.

5.9.2 Relations d’utilisation avec d’autres modules

Ce module sera en relation avec les modules “Package Profile”, “Package Wrapper” et “Preseed” mais également avec la base de données.

Pour fonctionner le module “Remaster” nécessite que certains fichiers soient préparés, pour cela nous avons créé un script qui s’appelle **Préparation Remaster** qui a été conçu, comme ceci :

5.10 Conception du script “Préparation Remaster”

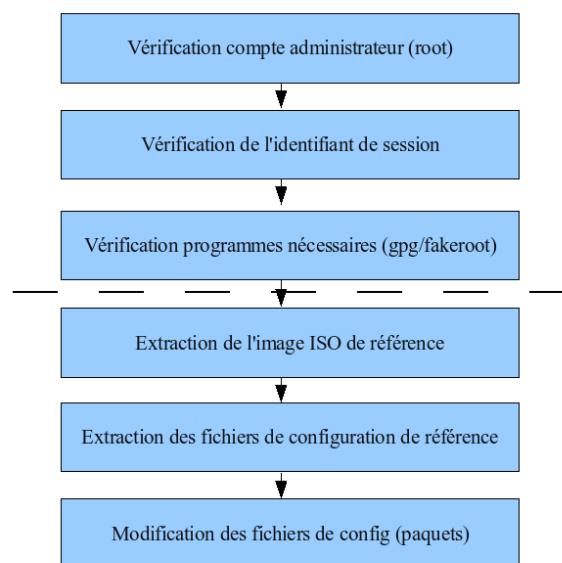


FIG. 11 – Schéma – Conception du script “Préparation Remaster”.

5.11 Conception du module "Remaster"

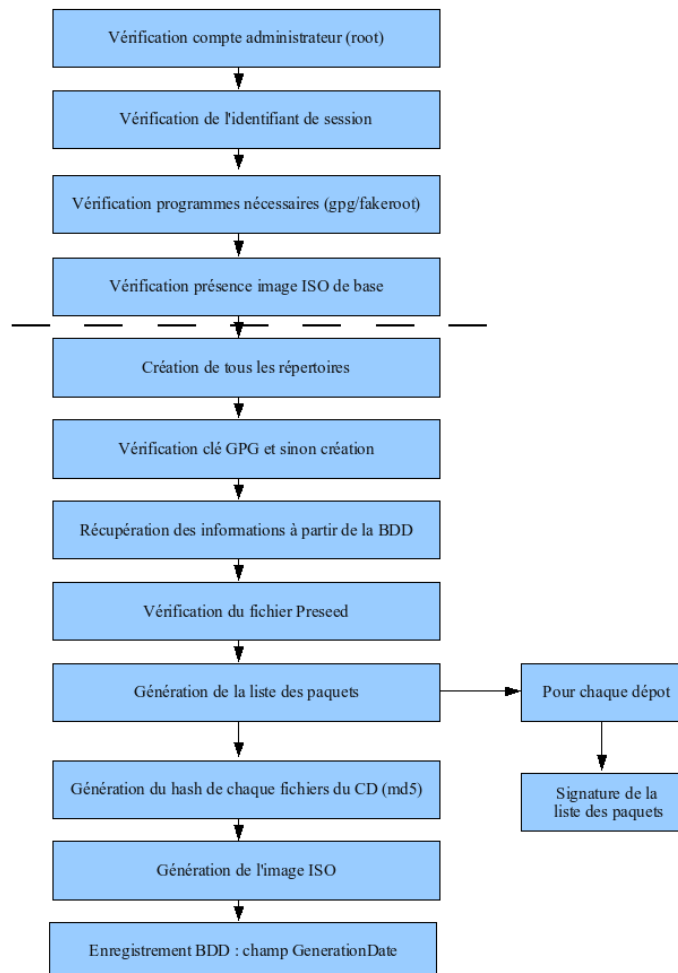


FIG. 12 – Schéma – Conception du module "Remaster".

5.11.1 Définitions de types

Ce paragraphe donne la liste des définitions de type. Dans le cas d'une conception orientée objets cette définition de type peut être unique et représenter les attributs de l'objet associé au module.

Nom variable	Type de données
IDSelection	int(8)
IsoName	varchar(75)
IsoFileName	varchar(100)
CreationDate	datetime
GenerationDate	datetime

5.11.2 Dépendances du modules

Pour fonctionner ce module a besoin des dépendances suivantes :

- StringIO
- gzip
- sysgzip
- MySQLdb

5.12 Module “Select”

5.12.1 Objectifs du module

Ce module est le **point d’entrée** de notre logiciel. En effet, c’est dans cette interface que l’utilisateur effectue la sélection et les différents choix de sa configuration.

Cette partie fait référence à la spécification d’interface Homme – Machine définie dans le dossier de spécifications. Ainsi, pour que l’utilisateur ne se **perde** pas dans tous les paramètres, nous allons afficher une page web **simple**.

On retrouve la division en 3 parties : la configuration du système de base (au niveau du Preseed), la configuration détaillée de paquets et enfin la sélection de profils d’applications.

Message d'accueil

Configuration du système de Base (au niveau du Preseed)

Nom utilisateur

Mot de passe

Choix langue

Configuration détaillée des paquets

- ▶ Personnalisation du navigateur internet
- ▶ Personnalisation de l'éditeur de texte
- ▶ Personnalisation du terminal
- ▶ Personnalisation du bureau
- ▶ Personnalisation des bureaux virtuels
- ▶ Personnalisation de l'interpréteur de commandes

Sélection des profils d'applications

- ✓ Extraction des archives de différents formats.
- ✓ Lecture de vidéos et sons en différents formats
- ✓ Environnement de Développement Intégré Java
- ✓ Environnement de Développement Intégré Python
- ✓ Utilitaire de montage vidéo
- ...

Valider la Sélection

Les cases à cocher sont représentées par des ✓.

FIG. 13 – IHM – Interface non détaillée de sélection de configuration de l’utilisateur .

5.12.2 Relations d'utilisation avec d'autres modules

Après les vérifications nécessaires, ce module sera en relation avec la base de données et plus particulièrement la table "Selection".

5.13 Conception de l'interface Homme – Machine

Pour concevoir cette spécification, nous utiliserons les technologies **Javascript** et **CSS**.

Message d'accueil

Configuration du système de Base (au niveau du Preseed)

Nom utilisateur

Mot de passe

Choix langue

Configuration détaillée des paquets

▼ Personnalisation du navigateur internet

- ✓ Utiliser l'ancien mode de fermeture des onglets .
- ✓ Cacher le bouton "Go".
- ✓ Forcer l'ouverture des liens "externes" dans un nouvel onglet.
- ✓ Désactiver la demande de fermeture des onglets.
- ✓ Activer le bouton du milieu.
- ✓ Activer le défilement avec le bouton du milieu.
- ✓ Désactiver le préchargement des sites.
- ✓ Désactiver le cache du disque.
- ✓ Désactiver l'historique de navigation.
- ✓ Désactiver les cookies.

- Personnalisation de l'éditeur de texte
- Personnalisation du terminal
- Personnalisation du bureau
- Personnalisation des bureaux virtuels
- Personnalisation de l'interpréteur de commandes

Sélection des profils d'applications

- ✓ Extraction des archives de différents formats.
- ✓ Lecture de vidéos et sons en différents formats
- ✓ Environnement de Développement Intégré Java
- ✓ Environnement de Développement Intégré Python
- ✓ Utilitaire de montage vidéo
- ...

Valider la Sélection

Les cases à cocher sont représentées par des √.

FIG. 14 – IHM – Interface détaillée de sélection de configuration de l'utilisateur .

Toutes les zones de choix de configuration masquées, représentées en **verte** sur l'IHM, seront délimitées par des balises HTML de type `</span>...</span>`, par défaut, elles seront masquées, nous utiliserons donc les attributs suivants : `style="visibility :hidden ;display :none"`.

Nous gérerons les événements, à l’aide du **Javascript**. Dès que l’utilisateur cliquera sur le titre du paquet à modifier par exemple “Personnalisation du navigateur internet” ou encore “Personnalisation de l’éditeur de texte” une fonction Javascript sera appelée.

L’événement déclenché appellera la fonction Javascript “toggleVisibilité” qui aura pour but d’**afficher** la zone si elle est masquée ou alors de la **cacher** si elle est affichée.

Pour afficher la zone, nous positionnerons les variables **Visibility** à “visible” et **Display** à “inline” de la zone sélectionnée et au contraire pour masquer la zone nous positionnerons les variables **Visibility** à “hidden” et **Display** à “none”.

5.14 Procédure de vérification de la sélection de l’utilisateur

Pour transmettre les informations du formulaire nous utilisons la méthode **POST**. La vérification va s’effectuer dans la page **PHP** appelée **insert.php**.

Pour récupérer la valeur des informations transmises, nous allons utiliser la fonction PHP **\$HTTP_POST_VARS**.

Les types suivants seront considérés comme **obligatoires** lors de l’enregistrement d’une nouvelle sélection dans la base de données et la page **insert.php** va donc vérifier qu’ils sont correctement renseignés :

- IDSelection ;
- preseed_lang ;
- preseed_network ;
- preseed_hostname ;
- preseed_domainname ;
- preseed_timezone ;
- preseed_user ;
- preseed_password ;
- IsoName ;
- IsoFileName.

5.15 Utilisation de fonctions PHP

Pour se connecter à la base de données et y réaliser toutes les opérations nécessaires, comme des sélections ou des insertions, nous utiliserons les fonctions **PHP/MySQL** suivantes :

- **mysql_connect()** : permet de se connecter à un serveur de base de données ;
- **mysql_select_db()** : permet de se connecter à une base de données ;
- **mysql_query()** : envoie une requête de sélection et d’ajout au SGBDR ;
- **mysql_fetch_row()** : réalise le traitement de chaque enregistrements du résultat d’une requête ;
- **mysql_fetch_array()** : récupère les informations sous forme de tableau PHP (array).

Dans le but de lancer la procédure de génération de l'image ISO, nous exécuterons alors le script de génération avec la fonction **popen()**, qui a pour but de lancer une commande au niveau non plus du serveur web mais du système d'exploitation.

La fonction **popen()** sera donc appelée avec en paramètre le **script Python** en charge de la génération et l'**identifiant** de l'utilisateur dans le but de récupérer les informations enregistrées dans la base de données.

5.15.1 Définitions de types

Les variables que nous utilisons pour ce module sont définies de la manière suivante :

Nom variable	Type de données
IDSelection	varchar(12)
preseed_lang	varchar(20)
preseed_network	tinyint(1)
preseed_hostname	varchar(20)
preseed_domainname	varchar(20)
preseed_timezone	varchar(45)
preseed_user	varchar(30)
preseed_password	varchar(30)
IsoName	varchar(200)
IsoFileName	varchar(200)

ISAIP
18, rue du 8 Mai 1945
49000 St Barthélémy d'Anjou



Dossier Protocoles de Tests

Développement d'une application de personnalisation de système
d'exploitation : "OS A La Carte"

1 Février 2007

Mohammed BADISS, Jérôme CHARLOT
2^{ème} année CPI
Promotion 2005 – 2007

Sous la responsabilité de :
M Alexandre LEPRIEULT ;



Table des matières

1	Traçabilité	3
2	Introduction	3
3	Rappels	3
3.1	Objectifs du projet	3
3.2	Procédé de fabrication	4
3.3	Les Fonctionnalités	5
3.4	Découpage de l’application en modules	6
3.5	Rôle de chaque module	6
4	Test d’intégration	7
4.1	Test du procédé de génération du DVD d’installation	7
4.2	Test de l’installation et de la sélection de l’utilisateur	8
5	Tests Unitaires	9
5.1	Test du module “Feed”	9
5.1.1	Test effectué	9
5.1.2	Matière en entrée	10
5.1.3	Résultat attendu	10
5.2	Test du module “Package profiles”	10
5.2.1	Test effectué	10
5.2.2	Matière en entrée	11
5.2.3	Résultat attendu	11
5.3	Test du module “Remaster”	11
5.3.1	Test effectué	11
5.3.2	Matière en entrée	12
5.3.3	Résultat attendu	12
5.4	Test du module “Select”	12
5.4.1	Test effectué	12
5.4.2	Matière en entrée	13
5.4.3	Résultat attendu	15
6	Résultats des tests effectués	15
6.1	Module “Feed”	15
6.2	Module “Packages Profiles”	15
6.3	Module “Remaster”	15
6.4	Module “Select”	15

Table des figures

1	Schéma – Déroulement du procédé de fabrication.	4
2	Schéma – Choix possibles de l'utilisateur.	5
3	Schéma – Découpage de l'application OS A La Carte en modules.	6
4	Schéma – Test de la génération du DVD d'installation.	7
5	Schéma – Test de l'installation et de la sélection de l'utilisateur.	8
6	Schéma – Test du module “Feed”.	9
7	Schéma – Exemple d'un paquet en entrée pour le fichier de test.	10
8	Schéma – Test du module “Feed”.	11
9	Schéma – Test du module “Remaster”.	11
10	Schéma – Test du module “Select”.	12

1 Traçabilité

Version	Date	Commentaires
1.00	06/01/07	Première version
1.30	03/02/07	Ajout de la partie Rappel
1.40	03/02/07	Ajout de la définition des objectifs des tests de non-regression
1.50	03/02/07	Ajout des résultats des tests effectués
2.00	04/02/07	Correction orthographe

2 Introduction

La méthode **Agile** que nous avons choisi s'appelle l'**Extreme Programming**, elle se caractérise par l'usage **intensif de tests**. On parle de tests de **non-régression** qui sont effectués sur le logiciel.

Les tests de **non-régression** ont pour but de vérifier que la **nouvelle** version du logiciel fonctionne de la même manière que la version **précédente**, et que les évolutions de la nouvelle version n'ont pas amené directement ou indirectement un mauvais fonctionnement.

Ainsi, dès que nous ajoutons de nouvelles modifications à notre programme, par exemple le support de nouveaux paquets ou l'ajout de fonctionnalités pour les paquets, nous **redéroulions** à chaque fois tous les tests de **A à Z**.

3 Rappels

Il est nécessaire d'effectuer quelques rappels, pour rappeler les parties importantes du déroulement du projet.

3.1 Objectifs du projet

Notre projet nommé “OS A La Carte” a pour but de **personnaliser** un système d'exploitation depuis une page Internet, plus particulièrement de générer une image du DVD d'installation d'une distribution Linux.

Cette image au format ISO est destinée à être gravée sur un DVD. Une fois gravée, l'utilisateur insère le DVD dans son lecteur de DVD puis amorce le processus d'installation. L'utilisateur installe alors ce système d'exploitation sur sa machine.

Notre travail repose donc entièrement sur la **modification** selon les demandes et les besoins du client de la distribution Linux nommée **Ubuntu**.

Cette **personnalisation** s'effectue à la base du système d'exploitation, tout d'abord au niveau des programmes qui sont couramment utilisés et mais également au niveau de l'installation d'applications additionnelles.

3.2 Procédé de fabrication

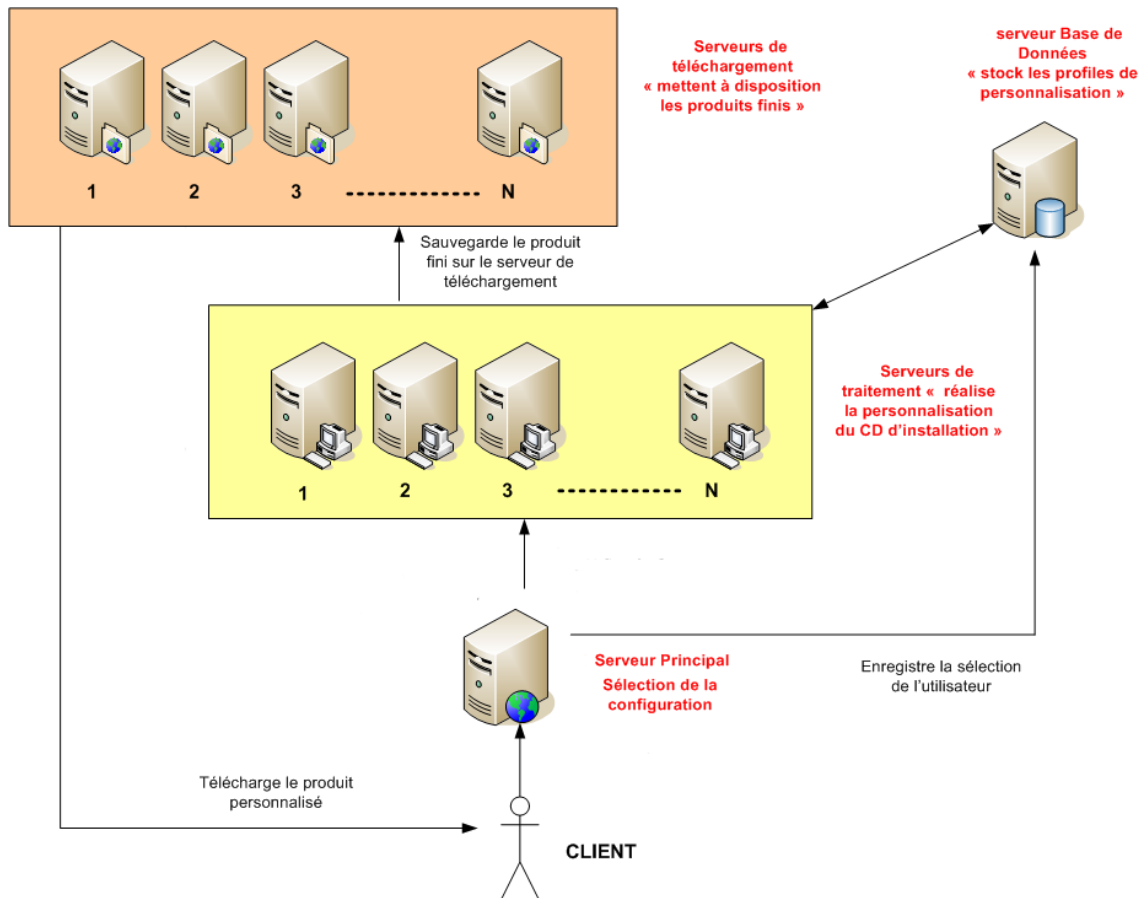


FIG. 1 – Schéma – Déroulement du procédé de fabrication.

Le procédé de fabrication se déroule en **trois** phases, qui sont les suivantes :

1. La première phase est la **sélection** de la configuration depuis notre interface Web. Cette opération est la seule effectuée par l'utilisateur. Le temps de réponse l'interface web dépend principalement de la bande passante du client, et de notre serveur. Mais avec les connexions actuelles, la technologie ADSL étant largement répandue, ce temps est **négligeable** (à peine quelques secondes).
2. Une fois la phase de personnalisation terminée. La phase de **génération** du DVD d'installation, réalisée par notre application, peut exiger environ **3** minutes sur le serveur dédié (un Pentium 4 ou équivalent).
3. Finalement, la phase de **téléchargement** de l'image du DVD est la plus **exigeante** au niveau temps, car comme la première phase, elle dépend de la bande passante que dispose le client mais surtout celle que nous allouons pour le serveur de téléchargement.

3.3 Les Fonctionnalités

Voici un schéma qui représente l'étendue des choix pour lesquels l'utilisateur peut personnaliser son système d'exploitation :

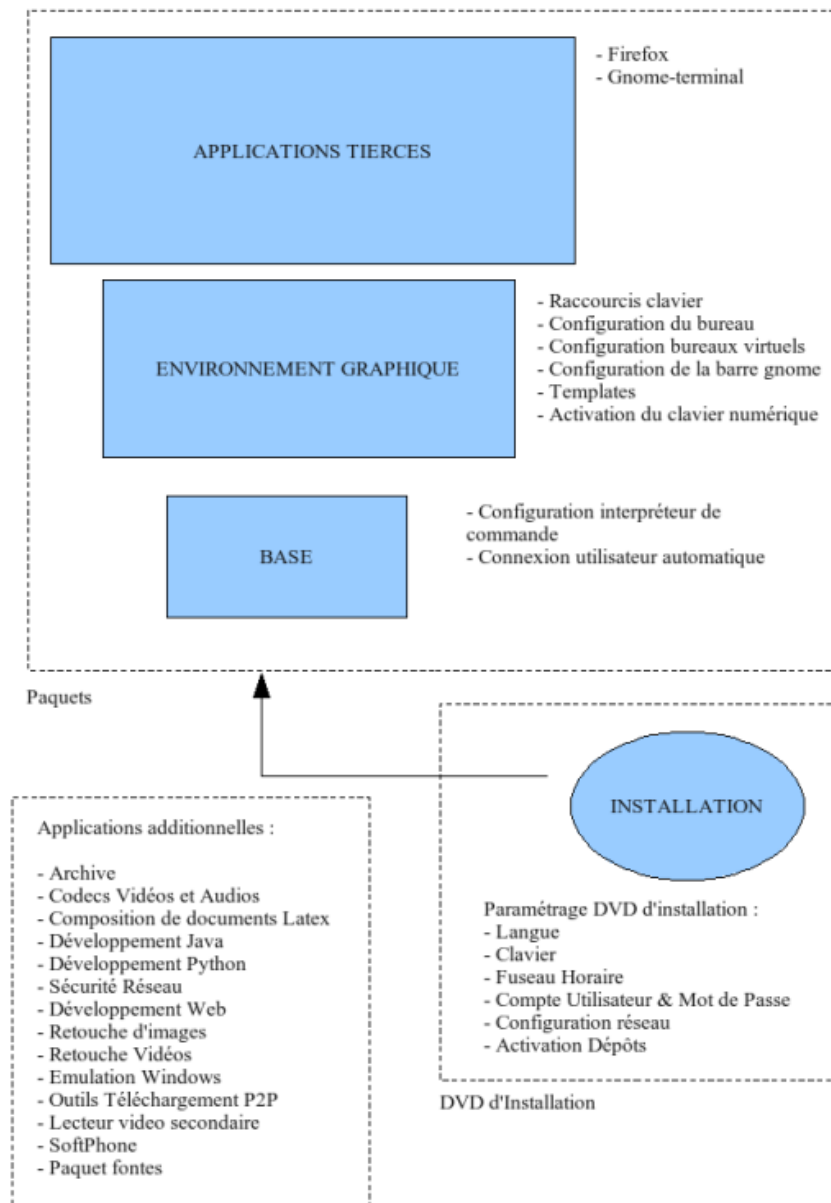


FIG. 2 – Schéma – Choix possibles de l'utilisateur.

L'utilisateur personnalise le DVD d'installation à **trois** niveaux différents.

Les trois niveaux sont :

1. Fonctionnalités au niveau du DVD d'installation.
2. Modifications des paquets les plus importants.
3. Choix des profils d'applications.

Ces différents niveaux sont **abstraits** et **transparents** pour l'utilisateur, ils correspondent à un découpage en modules de l'application. Ce découpage est détaillé dans la partie “*Découpage de l'application en modules*”.

3.4 Découpage de l'application en modules

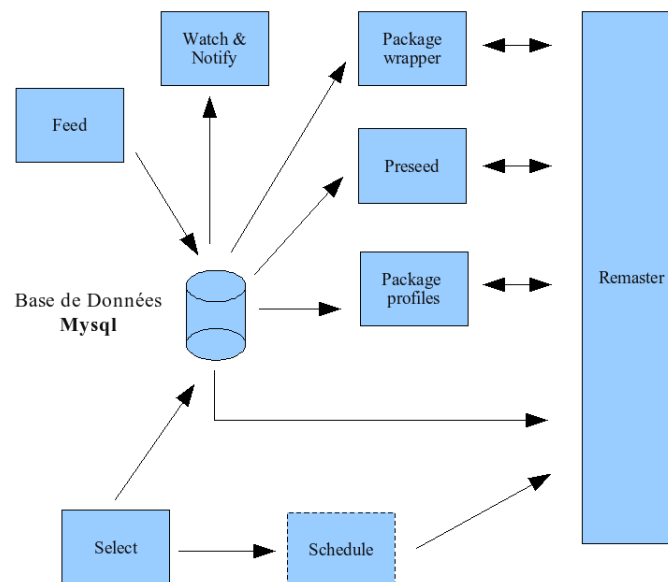


FIG. 3 – Schéma – Découpage de l'application OS A La Carte en modules.

3.5 Rôle de chaque module

- “**Feed**” prendra en compte le suivi des mises à jour proposées par les dépôts officiels d'Ubuntu (Réponse à la contrainte de *sécurité*) ;
- “**Watch and Notify**” aura pour rôle d'afficher les changements des dépôts ;
- “**Package Profiles**” s'occupera de la gestion des profils d'applications ;
- “**Package Wrapper**” gèrera les paquets, à savoir leur ouverture, leur modification, et finalement leur reconstruction ;
- “**Preseed**” concernera le paramétrage du CD d'installation ;
- “**Remaster**” aura pour tâche la création de l'image ISO en prenant en compte les modules “**Package Profile**”, “**Package Wrapper**” et “**Preseed**” ;

- “**Select**” sera l’interface où l’utilisateur sélectionnera ses choix ;
- “**Schedule**” gèrera les utilisateurs, avec une file d’attente, la sauvegarde des sélections des utilisateurs (Module Non Prioritaire).

4 Test d’intégration

Les tests d’intégration sont divisés en **deux** grandes parties. Tout d’abord, nous testons le programme de génération de l’image du DVD d’installation, puis nous installons le DVD produit dans un **environnement virtuel** et nous vérifions que la sélection de l’utilisateur a été réalisée.

4.1 Test du procédé de génération du DVD d’installation

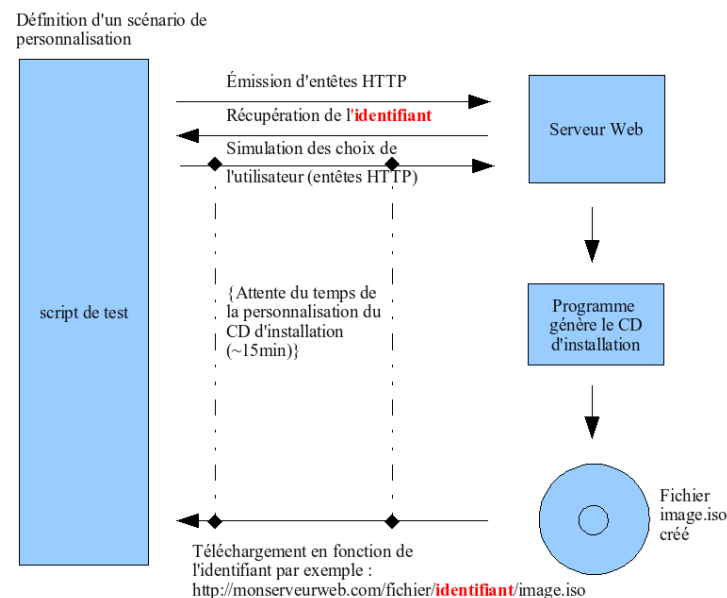


FIG. 4 – Schéma – Test de la génération du DVD d’installation.

Nous utiliserons pour cela les modules suivants en python :

- “python-mysqldb” qui permet de connecter à une base de données Mysql ;
- “urllib2” qui permet de manipuler les entêtes “HTTP” et donc les formulaires de type POST ;
- “httplib” qui permet de s’interfacer avec le protocole HTTP, notamment la négociation de la connexion ;
- “wget” pour télécharger l’image du DVD créée.

Une fois que l'image est produite, nous allons la télécharger et vérifier que celle-ci est pleinement fonctionnelle et correspond aux modifications souhaitées par l'utilisateur.

4.2 Test de l'installation et de la sélection de l'utilisateur

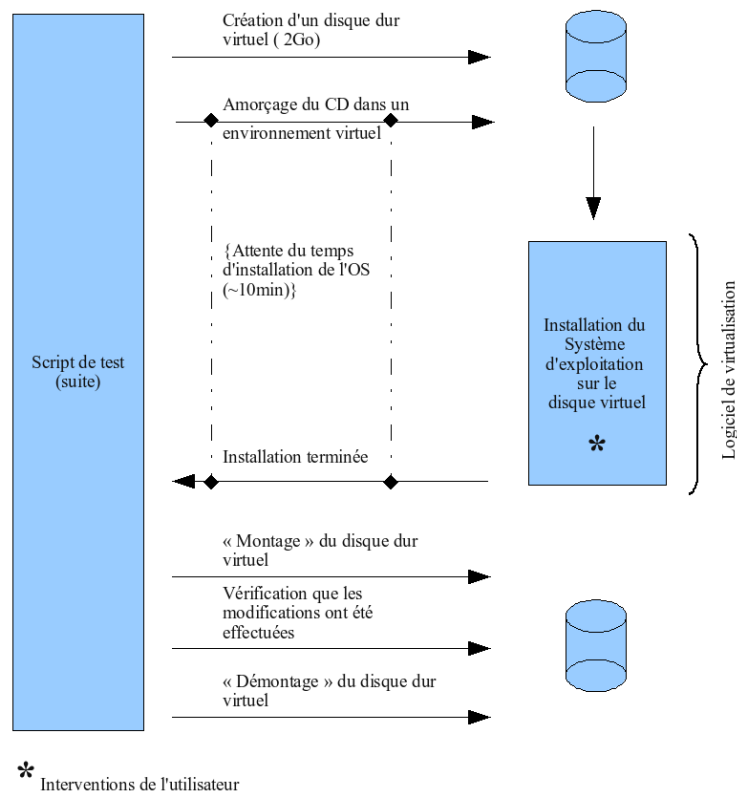


FIG. 5 – Schéma – Test de l'installation et de la sélection de l'utilisateur.

Pour exécuter le procédé d'installation nous utiliserons le logiciel **qemu** et son interface qui permet une accélération avec le noyau Linux : **kqemu**.

Ainsi les deux tests sont effectués **consécutivement** puisque le scénario en entrée, qui définit ce que l'utilisateur modifie, sera réutilisé à la fin du deuxième test, lorsque l'on "monte" le disque dur virtuel pour vérifier que toutes les modifications ont été prises en compte.

5 Tests Unitaires

5.1 Test du module “Feed”

5.1.1 Test effectué

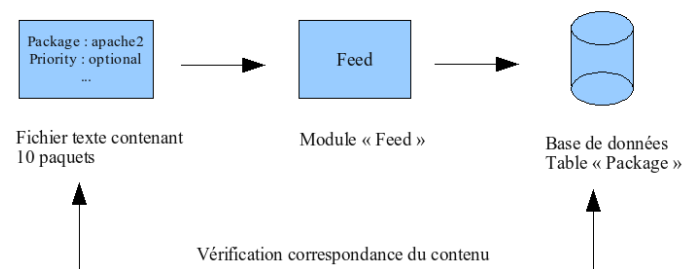


FIG. 6 – Schéma – Test du module “Feed”.

Nous créons un fichier texte contenant une liste de **10 paquets** de type différents (programmes, librairies,...) nous exécutons le module “Feed” avec en paramètre le nom du fichier à tester.

Ensuite nous nous connectons à la base de données et comparons les entrées fixées par le fichier texte et ce qui est inséré dans la base de donnée et notamment la table “Package”.

5.1.2 Matière en entrée

On créer un fichier contenant 10 paquets ayant comme structure la suivante :

```
Package: apache2
Priority: optional
Section: web
Installed-Size: 80
Maintainer: Ubuntu Core Developers <ubuntu-devel@lists.ubuntu.com>
Original-Maintainer: Debian Apache Maintainers <debian-apache@lists.debian.org>
Architecture: i386
Version: 2.0.55-4ubuntu4
Depends: apache2-mpm-worker (= 2.0.55-4ubuntu4) | apache2-mpm-prefork (= 2.0.55-
| apache2-mpm-perchild (= 2.0.55-4ubuntu4)
Filename: pool/main/a/apache2/apache2_2.0.55-4ubuntu4_i386.deb
Size: 35976
MD5sum: b41e510af15a616746930c7ba86362c2
SHA1: be0399c61f2b3e3a27a0b1f2f558bc7218b7f7bc
SHA256: 2d1659ecb4512b257c933a2c9842f8a416641f939000f7d105ef9fa30c167a3b
Description: next generation, scalable, extendable web server
 Apache v2 is the next generation of the omnipresent Apache web server. This
 version - a total rewrite - introduces many new improvements, such as
  threading, a new API, IPv6 support, request/response filtering, and more.
Bugs: mailto:ubuntu-users@lists.ubuntu.com
Origin: Ubuntu
Task: lamp-server
```

FIG. 7 – Schéma – Exemple d’un paquet en entrée pour le fichier de test.

5.1.3 Résultat attendu

Les valeurs qui sont structurées dans le fichier texte doivent avoir le même contenu dans la base de données.

5.2 Test du module “Package profiles”

5.2.1 Test effectué

On lance le module “Package Profile” avec comme paramètre un **identifiant** de sélection qui doit être existant. Cet identifiant fait référence à une sélection de profils d’applications. Il doit y avoir correspondance entre la sélection de la base de données (table “Selection”) et les fichiers qui sont copiés dans le répertoire de génération du DVD d’installation.

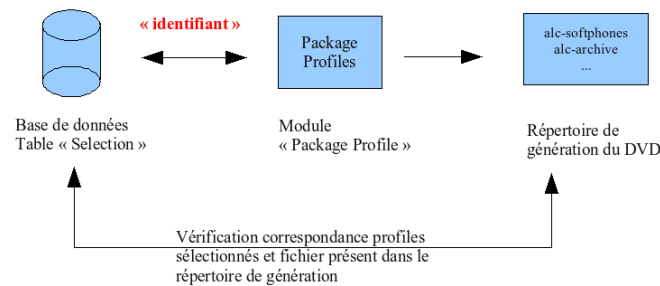


FIG. 8 – Schéma – Test du module “Feed”.

5.2.2 Matière en entrée

La matière en entrée sont des enregistrements dans la base données ainsi on peut avoir les enregistrements suivants :

```
alc-softphones
alc-archives
alc-edi
alc-edi-java
```

5.2.3 Résultat attendu

Selon les profils d’applications sélectionnés, les paquets correspondants doivent être copiés dans le répertoire de génération du DVD d’installation.

5.3 Test du module “Remaster”

5.3.1 Test effectué

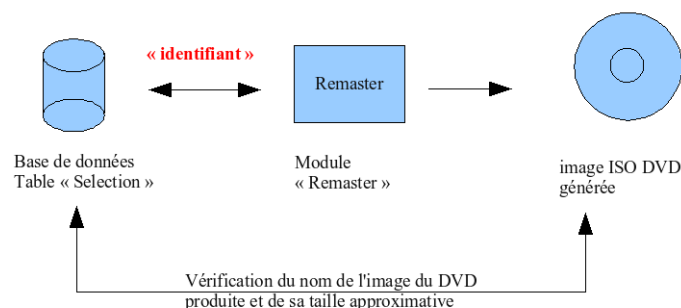


FIG. 9 – Schéma – Test du module “Remaster”.

On lance le module “Remaster” avec comme paramètre un **identifiant** de sélection qui doit être existant. Cet identifiant fait référence à une sélection de profils d’applications. Dans la base de données et

plus particulièrement la table “Selection” figure des champs concernant qui font références au **nom de l’image** et au **nom du fichier** qui aura pour extension “.iso”. Ainsi ces informations doivent être cohérentes.

De plus, l’utilisateur sélectionne un certains nombre de profils d’applications et ils s’ajoutent à la taille du DVD. On peut ainsi vérifier la **taille approximative** de l’image ISO du DVD.

5.3.2 Matière en entrée

Avec l’identifiant passé en paramètre, nous disposons dans la base de données des champs “IsoName” et “IsoNameFile” qui correspondent au nom de l’image générée.

De plus l’utilisateur sélectionne des profils d’applications et ces profils occupent une taille fixe or on sait que le DVD occupe par défaut 700 Mo, ainsi on calcule une estimation de la taille de la future image ISO.

5.3.3 Résultat attendu

Les noms des fichiers et ceux qui sont inscrits dans la base de données doivent correspondre et l’estimation de la taille du fichier image ISO doit correspondre avec le calcul de son estimation.

5.4 Test du module “Select”

5.4.1 Test effectué

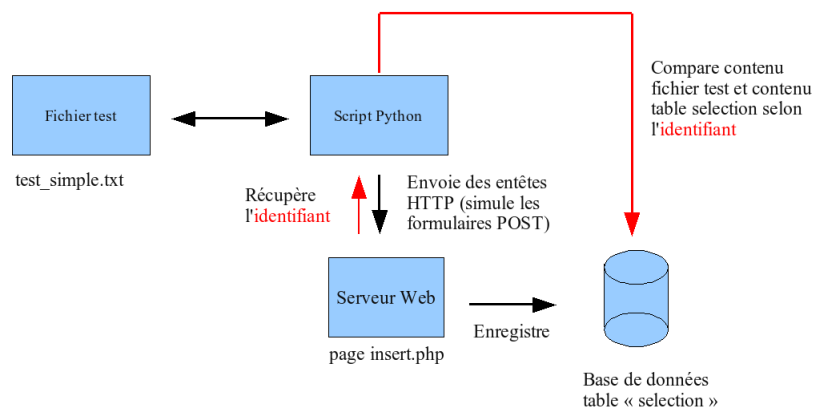


FIG. 10 – Schéma – Test du module “Select”.

Objectif :

L’objectif de ce test est de simuler les choix de l’utilisateur.

Ce test est important car il s’agit du “point d’entrée” de tout le processus de personnalisation de l’image

du DVD.

Ainsi le test se déroule en **10** étapes :

1. Lancement du programme de test “select_module_test.py”.
2. Vérification de la présence du paramètre.
3. Vérification que le paramètre fourni est un nom de fichier.
4. Vérification du contenu du fichier de test.
5. Connexion au formulaire de personnalisation (avec comme paramètre *test=1*).
6. Récupération de l'**identifiant** attribué par le serveur, celui-ci est **unique**.
7. Fabrication et émission des entêtes HTTP sur la page **insert.php**.
8. Récupération du code de retour et du contenu de la page du serveur web.
9. Connexion à la base de données et requête sur la table “**selection**” avec comme argument l'**identifiant** fourni depuis le premier échange avec le serveur web.
10. Comparaison entre le contenu fourni en fichier d'entrée et les informations de la base de données.

5.4.2 Matière en entrée

En matière d'entrée, nous avons un fichier texte qui contient les informations suivantes :

```
IsoName=Nom interne
IsoFileName=Monfichier.iso
preseed_lang=fr-FR
preseed_network=1
preseed_hostname=bruce
preseed_domainname=bsheep.org
preseed_timezone=Europe/London
preseed_user=bsheep
preseed_password=monmotdepasse
preseed_repo=1
preseed_network=1
preseed_ip=172.16.84.34
preseed_dns=172.16.84.1
preseed_netmask=255.255.0.0
preseed_gateway=172.16.84.1
firefox_oldclose=1
firefox_go=1
firefox_forceexternalpage=1
firefox_askclose=1
firefox_middlebutton=1
firefox_smoothscrolling=1
firefox_prefetch=1
firefox_cachedisk=1
firefox_navigationhistory=1
firefox_cookies=1
firefox_sessioncookies=1
```

```
firefox_downloadhistory=1
firefox_javaplugin=1
firefox_flashplugin=1
firefox_videoplugin=1
firefox_adblockplugin=1
gedit_indent=1
gedit_highlight=1
gedit_font=1
gedit_linenumber=1
gedit_autosave=1
gedit_savecopy=1
gedit_spaceindent=1
gedit_plugin=spell,time
terminal_keyboardshortcut=1
terminal_close=CTRL + Q
terminal_open=CTRL + T
terminal_right=CTRL + D
terminal_left=CTRL + F
nautilus_computericon=1
nautilus_homeicon=1
nautilus_trashicon=1
nautilus_volumesicon=1
nautilus_networkicon=1
nautilus_tighterlayout=1
nautilus_imagescript=1
nautilus_documentmodels=1
metacity_keyboardshortcut=1
metacity_fileexplorer=TOUCHE Windows + A
metacity_texteditor=TOUCHE Windows + B
metacity_wordprocessor=TOUCHE Windows + C
metacity_internetbrowser=TOUCHE Windows + D
metacity_mailapp=TOUCHE Windows + E
metacity_terminal=TOUCHE Windows + F
metacity_numworkspace=4
metacity_workspacename1=Communication
metacity_workspacename2=Internet
metacity_workspacename3=Code
metacity_workspacename4=Musique
metacity_workspacename5=Film
metacity_workspacename6=Divers
metacity_workspacename7=Divers2
metacity_workspacename8=Divers3
metacity_workspacename9=Divers4
metacity_workspacename10=Divers5
bash_completion=1
bash_promptcoloration=1
bash_alias=
alc-archive=1
```

```

alc-codecs-xine=1
alc-latex=1
alc-edi=1
alc-edi-java=1
alc-edi-python=1
alc-network-tools=1
alc-web-development=1
alc-graphic-tools=1
alc-video-edition=1
alc-windows-api=1
alc-download-soft=1
alc-videoplayer=1
alc-softphones=1
alc-fonts=1

```

5.4.3 Résultat attendu

Le résultat attendu est un enregistrement dans la base de données, ou les informations dans le fichier texte et les champs remplis dans la base de données doivent correspondre.

6 Résultats des tests effectués

6.1 Module "Feed"

Test	Résultat attendu	Résultat obtenu	Date	Responsable
Vérification Fichier entrée et BDD	Cohérence Fichier et BDD	Cohérence Fichier et BDD	03/02/07	M BADISS

6.2 Module "Packages Profiles"

Test	Résultat attendu	Résultat obtenu	Date	Responsable
Vérification enregistrement BDD paquets installés	Cohérence enregistrement BDD paquets installés	Cohérence enregistrement BDD paquets installés	03/02/07	J CHARLOT

6.3 Module "Remaster"

Test	Résultat attendu	Résultat obtenu	Date	Responsable
Vérification enregistrement BDD fichier ISO	Cohérence enregistrement BDD fichier ISO	Cohérence enregistrement BDD fichier ISO	03/02/07	J CHARLOT

6.4 Module "Select"

Test	Résultat attendu	Résultat obtenu	Date	Responsable
Vérification Fichier entrée et BDD	Cohérence Fichier et BDD	Cohérence Fichier et BDD	03/02/07	M BADISS

ISAIP
18, rue du 8 Mai 1945
49000 St Barthélémy d'Anjou



Journal de Bord

Développement d'une application de personnalisation de système
d'exploitation : "OS A La Carte"

1 Février 2007

Mohammed BADISS, Jérôme CHARLOT
2^{ème} année CPI
Promotion 2005 – 2007

Sous la responsabilité de :
M Alexandre LEPRIEULT ;



Table des matières

1	Décisions Importantes.	2
2	Chronologie des événements.	2
3	État d'avancement.	2
4	Compte rendu – Réunion du 09 Novembre 2006.	3
5	Compte rendu – Réunion du 30 Novembre 2006.	4
6	Compte rendu – Réunion du 20 Décembre 2006.	5
7	Compte rendu – Réunion du 2 Février 2007.	6

1 Décisions Importantes.

Nous avons décidé de déposer un dossier auprès de SourceForge pour héberger le projet.
Nous avons décidé de découper le projet en 7 sous modules.

2 Chronologie des événements.

9 Novembre Réunion de validation du projet.
15 Novembre Demande d'hébergement auprès de SourceForge.
22 Novembre Le projet est accepté par SourceForge.
30 Novembre Réunion Cahier des charges et Présentation du planning prévisionnel.
11 Décembre Création du site de présentation du projet (<http://osalcproject.sourceforge.net/index1.html>)
15 Décembre Début du développement en équipe (utilisation outils)
20 Décembre Début de réalisation du dossier Gestion Financière (simulation de mise en activité de notre projet)
20 Décembre Réunion de présentation du dossier de spécifications fonctionnelles
17 Janvier Réunion de présentation du dossier de conception et protocoles de tests

3 État d'avancement.

4 Compte rendu – Réunion du 09 Novembre 2006.

Professeurs présents :

- M Alexandre *LEPRIEULT*
- Mlle Amandine *DUFFOUX*
- M Redouane *DJELOUAH*

Étudiants présents :

- M Mohammed *BADISS*
- M Jérôme *CHARLOT*

Compte rendu :

Lors de la réunion de validation de sujet qui s’est déroulée le 09 novembre 2006, nous avons exposé le sujet de notre projet technique nommé “Os A La Carte”.

L’objectif du projet est la personnalisation d’un système d’exploitation depuis une interface web. Il vise à simplifier la configuration du système d’exploitation Linux pour les utilisateurs débutants principalement.

M *DJELOUAH* a exprimé sa crainte concernant la taille de notre projet. Mais nous avons répondu que nous avons prévu, dès le début du projet, l’utilisation d’un **prototype** pour surmonter les points critiques éventuels lors de son élaboration.

Le projet a été accepté après explication de son contenu et de son intérêt.

5 Compte rendu – Réunion du 30 Novembre 2006.

Professeur présent :

– M Alexandre *LEPRIEULT*

Étudiants présents :

– M Mohammed *BADISS*

– M Jérôme *CHARLOT*

Compte rendu :

Nous exposons à notre tuteur M *LEPRIEULT* nos différents dossiers, en lui faisant part de deux plannings prévisionnels, le premier qui prend en compte les phases du projet, analyse, spécifications, conception et un second qui prend en compte une division en modules du projet.

Nous lui expliquons que nous avons des difficultés concernant l’élaboration des dossiers. Notre tuteur nous explique alors que chez *NEC-Packard Bell*, ils ont déjà travaillé sur un projet équivalent. Selon lui, pour ce type de projet, il y aurait vraisemblablement un seul dossier qui contiendrait à la fois cahier des charges, dossier de spécifications et enfin dossier de conception.

Il nous rappelle ensuite que le dossier de spécifications, c’est le dossier qui exprime les souhaits du client et le dossier de conception, explique la manière dont nous répondons aux besoins.

Enfin la réunion s’achève en nous disant qu’il ne faut pas hésiter sur la quantité de schémas, dans le but de rendre les mécanismes et interactions plus claires.

6 Compte rendu – Réunion du 20 Décembre 2006.

Professeur présent :

- M Alexandre *LEPRIEULT*

Étudiants présents :

- M Mohammed *BADISS*
- M Jérôme *CHARLOT*

Compte rendu :

Lors de cette réunion, nous avons présenté les idées majeures de notre dossier de spécifications, à savoir la définition détaillée de l’architecture du projet et de l’analyse des fonctionnalités pour chacun des modules.

Cette réunion a été avancée, c’est pourquoi nous ne fournissons pas tous les documents complètement achevés.

Notre tuteur nous conseille de détailler d’avantage la définition des fonctionnalités. De plus il nous met à disposition un contact pour pouvoir par la suite continuer le projet tout en étant sponsorisé par une entreprise.

7 Compte rendu – Réunion du 2 Février 2007.

Professeur présent :

– M Alexandre *LEPRIEULT*

Étudiants présents :

– M Mohammed *BADISS*

– M Jérôme *CHARLOT*

Compte rendu :

Lors de cette réunion, nous avons présenté le dossier de spécifications, M Alexandre *LEPRIEULT* nous fait remarquer qu’il manque une partie sur l’interface Homme Machine concernant le module de choix de la sélection de l’utilisateur (“Select”). De plus il nous rappelle qu’il s’agit d’un projet **réel**.

Comme nous utilisons une méthode **Agile** (plus particulièrement l’**“Extreme Programming”**), nous lui présentons le dossier des protocoles de tests, nous lui faisons part que nous avons donc commencé par les tests et que nous avons donc réalisé tout un programme pour les réaliser de manière systématique.

Nous lui présentons enfin le dossier de conception, notre tuteur nous indique qu’il faudrait d’avantage détailler le coeur du programme, pour que le développeur en charge de l’action puisse travailler en autonomie.

De plus, il nous conseille d’ajouter de la documentation supplémentaire, sous forme de rappel dans les dossiers de protocole de tests et de conception, pour rappeler succinctement l’objectif général, les différentes fonctionnalités, le découpage en modules et enfin le rôle de chaque module.

Et enfin, il nous fait noter que tous les dossiers doivent avoir une rubrique de “Traçabilité”, qui concerne l’historique du document.

ISAIP
18, rue du 8 Mai 1945
49000 St Barthélémy d'Anjou



Dossier Économique

Développement d'une application de personnalisation de système
d'exploitation : "OS A La Carte"
2 Février 2007

Mohammed BADISS, Jérôme CHARLOT
2^{ème} année CPI
Promotion 2005 – 2007

Sous la responsabilité de :
M Pierre LAPLAZE ;
M Alexandre LEPRIEULT ;



Table des matières

1	Objectif de ce document	3
2	Présentation du projet	3
3	Présentation du procédé de fabrication	4
4	Notre besoin	5
4.1	Architecture matérielle	5
4.2	Architecture logicielle	6
4.3	Critères qui influent sur la Qualité du Service.	6
5	Recherche d’un hébergeur dédié	7
5.1	Offre OVH “Kimsufi” [17]	7
5.2	Offre Free “Dédibox” [16]	7
5.3	Choix d’une offre	7
6	Études des concurrents	8
6.1	Comparatif des distributions Linux les plus importantes	8
7	Études des coûts	9
8	Apports de notre projet	9

Table des figures

1	Tableau – Estimation du temps de téléchargement du DVD d’installation selon les différents types de connexion.	4
2	Schéma – Ajout de machines pour répondre à la demande de l’évolutivité du service (du nombre de demandes).	5
3	Tableau – Composants logiciels de l’application OS A La Carte.	6
4	Tableau – Spécifications techniques de l’offre “Kimsufi”.	7
5	Tableau – Spécifications techniques de l’offre “Dédibox”.	7
6	Tableau – Récapitulatif des coûts engendrés par le projet “OS A La Carte” pour une année.	9

1 Objectif de ce document

L’objectif de ce dossier est calculer les coûts relatifs à la simulation d’une mise en **production** de notre projet pour un certain nombre de clients. Cette opération consiste à **étudier** les besoins nécessaires et à **mettre en place** les architectures logicielles et matérielles pour le lancement du service.

2 Présentation du projet

Notre projet a pour rôle de rendre un système d’exploitation **pleinement** utilisable. Tout d’abord, il convient de détailler toutes les étapes nécessaires pour arriver à cette **finalité**.

Lorsqu’une personne souhaite utiliser son ordinateur, elle commence par **installer** son système d’exploitation. Pour cela, elle insère le média d’installation, il s’agit généralement d’un CD ou d’un DVD, elle démarre sur le média (“boot” sur le média) après le BIOS. Elle installe alors le système d’exploitation, une fois cette étape achevée, elle installe ensuite les **périphériques** avec des pilotes. Elle continue par l’installation des **applications** selon ses besoins et enfin elle termine par la **configuration** de celles-ci.

C’est dans ce contexte que notre projet prend naissance. En effet, notre projet a pour rôle de prendre en compte toutes ces actions et de construire un système d’exploitation dit “**clé en main**” ou **personnalisé**, pour qu’il corresponde le plus possible aux exigences et besoins de l’utilisateur.

Notre système d’exploitation se base sur une distribution Linux nommée **Ubuntu** [13]. Les programmes sont distribués sous forme de paquets qui sont disponibles dans des dépôts.

Notre projet s’appelle “OS A La Carte”, à la carte, faisant référence aux restaurants, où le client **choisit** les éléments de son menu.

3 Présentation du procédé de fabrication

Le procédé de fabrication se déroule en **trois** phases, qui sont les suivantes :

1. La première phase est la **sélection** de la configuration depuis notre interface Web. Cette opération est la seule effectuée par l'utilisateur. Le temps de réponse l'interface web dépend principalement de la bande passante du client, et de notre serveur. Mais avec les connexions actuelles, la technologie ADSL étant largement répandue, ce temps est **négligeable** (à peine quelques secondes). Cependant cette phase peut être influencée également par le nombre d'utilisateurs.
2. Une fois la phase de personnalisation terminée. La phase de **génération** du DVD d'installation, réalisée par notre application, peut exiger environ **3** minutes sur le serveur dédié (un Pentium 4 ou équivalent).
3. Finalement, la phase de **téléchargement** de l'image du DVD est la plus **exigeante** au niveau temps, car comme la première phase, elle dépend de la bande passante que dispose le client mais surtout celle que nous allouons pour le serveur de téléchargement. Ainsi dans des conditions idéales, à savoir si nos serveurs peuvent suivre la charge de la bande passante (que nous disposons de suffisamment de bande passante), on peut dresser un tableau du temps d'attente nécessaire à l'utilisateur pour télécharger l'image du DVD d'installation :

Type de connexion	Débit effectif descendant	Temps de téléchargement
1 Mega	130 Ko/s	5h12min
8 Mega	1 Mo/s	18min
20 Mega	2,5 Mo/s	8min

FIG. 1 – Tableau – Estimation du temps de téléchargement du DVD d'installation selon les différents types de connexion.

Ainsi le temps d'attente total pour l'utilisateur ayant une très bonne connexion sera de **13min** alors que pour un utilisateur ayant une connexion dite "moyenne" sera de **5h17min** .

4 Notre besoin

Le schéma suivant présente l'organisation des différents serveurs :

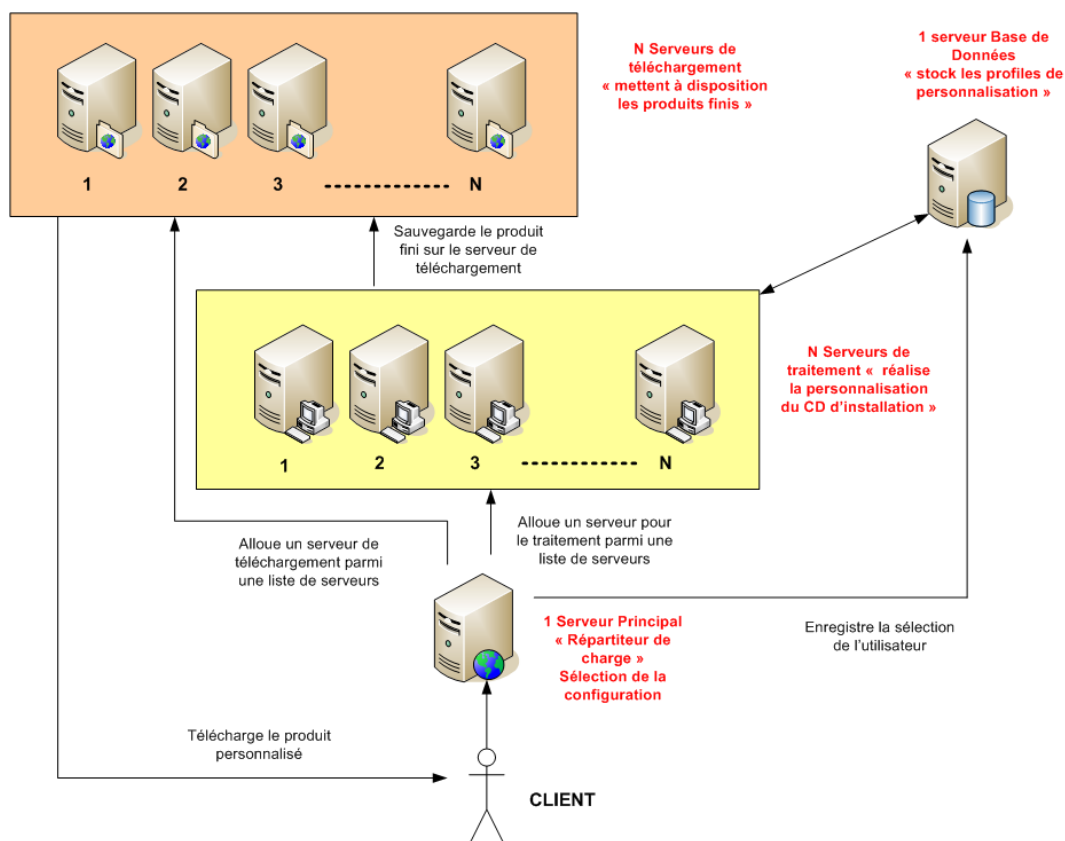


FIG. 2 – Schéma – Ajout de machines pour répondre à la demande de l'évolutivité du service (du nombre de demandes).

4.1 Architecture matérielle

Il nous faut un **répartiteur de charge** ("LoadBalancer"), une machine serveur de base de données (ou les profiles seront stockés), une machine qui réalise le **traitement**, et une machine qui permet la mise à disposition des fichiers images.

Avec le nombre de client qui augmente il faudra ajouter de nouveaux serveurs, qui seront de deux types : le serveur de traitement et celui de téléchargement.

Note : ici toutes les machines sont équivalentes et donc de même capacité.

4.2 Architecture logicielle

Notre architecture fait état de **quatre** types de serveurs qui sont :

1. Un serveur *web*, ou le client réalise sa sélection ;
2. Un ou plusieurs serveurs qui réalisent le *traitement* (génération de l'image du DVD) ;
3. Un serveur de *téléchargement* qui met à disposition l'image au format ISO au client ;
4. Un serveur de *base de données*, qui stocke les informations de l'utilisateur.

Voici les composants logiciels :

Type de Serveur	Programme	Version
Web	Apache	2.0.55
Web	PHP5	5.1.6
Web	php5-mysql	5.1.6
Téléchargement	Apache	2.0.55
Base de données	Mysql	5.0.24
Traitement	Python	2.4.4
Traitement	Python-mysqldb	1.2.1

FIG. 3 – Tableau – Composants logiciels de l'application OS A La Carte.

4.3 Critères qui influent sur la Qualité du Service.

Ainsi, dans le but d'améliorer le temps d'attente de l'utilisateur, on peut optimiser le service à **deux** niveaux. Si on augmente le nombre de machines de **traitement**, l'utilisateur attendra moins longtemps lors du procédé de génération et si on augmente le nombre de machine pour le **téléchargement**, ce qui aura pour rôle d'augmenter la bande passante et de diminuer le temps de téléchargement de l'image ISO pour l'utilisateur.

Ainsi pour lancer le projet, nous choisissons de privilégier le facteur de la **bande passante** et avons besoin des serveurs suivants : 1 Serveur de base de données, 1 serveur de traitement, 2 serveurs de téléchargement et enfin 1 serveur de sélection (qui joue le rôle de répartiteur de charge). Il nous faut donc en tout **5** serveurs.

5 Recherche d’un hébergeur dédié

Il existe deux offres possibles qui proposent d’héberger un système d’exploitation complet.

5.1 Offre OVH “Kimsufi” [17]

Critères	Détail
Processeur	Intel Celeron 2,00 GHZ
Architecture	32 bits
Mémoire vive	256 Mo DDR
Disque Dur	160 GO
Débit connexion (envoi)	100 Mbps
Débit connexion (réception)	100 Mbps
Nombre d’adresse IP	1
Prix (HT)	19,99 €
Frais d’installation	Gratuit

FIG. 4 – Tableau – Spécifications techniques de l’offre “Kimsufi”.

5.2 Offre Free “Dédibox” [16]

Critères	Détail
Processeur	VIA C7 2,00 GHZ
Architecture	32 bits
Mémoire vive	1 Go DDR
Disque Dur	160 GO
Débit connexion (envoi)	100 Mbps
Débit connexion (réception)	100 Mbps
Nombre d’adresse IP	1
Prix (HT)	29,99 €
Frais d’installation	Gratuit

FIG. 5 – Tableau – Spécifications techniques de l’offre “Dédibox”.

5.3 Choix d’une offre

Les deux offres sont donc très proches, elles sont **sans engagement** de durée. Cependant l’offre “dédibox”[16] a un processeur peu puissant et un prix très supérieur. Nous choisissons donc l’offre de OVH appelée “Kimsufi”[17].

Cependant si nous choisissons cette offre nous aurons à nous adapter à une distribution Linux qui n’est pas **Ubuntu**[13] mais **Debian**, ce qui provoquera de petits changements dans notre application.

6 Études des concurrents

6.1 Comparatif des distributions Linux les plus importantes

Il est important de faire un état des lieux des distributions Linux existantes pour savoir dans quel marché notre produit va se situer.

Critères de comparaison	Mandriva	Fedora	SuSE	Ubuntu
Version actuelle	2007.0	6	10.2	6.10
Date de sortie	octobre 2006	octobre 2006	décembre 2006	octobre 2006
Fréquences de sortie*	annuelle	semestrielle	semestrielle	semestrielle
Nombre de programmes supportés	10 000	15 000	12 000	22 000
Médias	1 DVD/ 3 CD	1 DVD/ 5 CD	1 DVD/4 CD	1 DVD/1 CD
Cibles	Particuliers –	Particuliers Entreprises	Particuliers Entreprises	Particuliers –
Support proposé	Téléphone[20] Forum wiki Courriel	Wiki Forum – –	Téléphone Forum Wiki Courriel	Wiki Forum – –
Prix	75 €[21]	Gratuit	47 €[22]	Gratuit
Parts de marché[19]	9%	14%	20%	22%

* Date ou l'éditeur sort une nouvelle version de sa distribution.

Note : Nous n'offrons que peu de support par rapport aux éditeurs **Mandriva** ou **SuSE**, cette remarque serait à prendre en compte si nous devions définir le prix de vente du produit fini.

7 Études des coûts

Dénomination	Prix unitaire	Quantité	Prix total (HT)
Encadrement Projet M Leprieult	80 €/h	3h	240 €
Encadrement Projet M Laplaze	80 €/h	1h	80 €
Achat d'un nom de domaine	15 €/an	1 an	15 €
Location machine pour proposer le service	240€/an [17]	5	1 200 €
Paieement développeur (conception application)	–	100 h	–
TOTAL (HT)	–	–	1 535 €

FIG. 6 – Tableau – Récapitulatif des coûts engendrés par le projet “OS A La Carte” pour une année.

8 Apports de notre projet

- Pour l’entreprise tout d’abord.

Selon un article de **01net** (CF Annexe “Article de presse - 01net - “Peut-on encore se passer d’Eclipse ?””), les entreprises n’ont pas de distributeur pour l’environnement de développement (**EDI**) qu’elles utilisent.

Ainsi, elles sont donc obligées, soit de maintenir eux même les versions de l’environnement de développement, soit de laisser les développeurs se débrouiller par eux-même.

Notre application s’inscrit dans ce contexte et propose ainsi des modules répondants aux différents **besoins**. Par exemple on trouvera un module **Eclipse général**, un module **Eclipse UML** pour créer de la documentation en norme UML, un module **Eclipse Unit** pour réaliser les tests unitaires.

- Pour le particulier, un système “**clé en main**” permet d’avoir un système complètement **fonctionnel**.

Selon les tâches qu’il souhaitera réaliser, nous lui mettons déjà à disposition des **profils d’applications**.

Nous proposons une solution de “**montage vidéo**”, qui regroupera tous les utilitaires nécessaires à cette tâche comme l’acquisition de la vidéo, programme de montage vidéo et programme de création de DVD.

L’utilisateur peut également avoir une solution “**multimédia**”, qui prend en compte plus de 200 formats vidéos et 50 formats audios.

De plus, si cette personne est un **développeur**, nous lui proposerons d’installer tous les outils nécessaires pour la programmation. Ainsi nous proposons déjà, des profils “Développement Web” (PHP), “Développement Python” et “Développement Java”.

ISAIP
18, rue du 8 Mai 1945
49000 St Barthélémy d'Anjou



Documentation Officielle : Custom Install CD

Développement d'une application de personnalisation de système
d'exploitation : "OS A La Carte"

26 Novembre 2006

Mohammed BADISS, Jérôme CHARLOT
2^{ème} année CPI
Promotion 2005 – 2007

Sous la responsabilité de :
M Alexandre LEPRIEULT ;



Table des matières

1	Preface	2
2	Introduction	2
3	Copy the CD to your hard drive	2
4	Modify installer behaviour using a Preseed file	2
4.1	Changing isolinux.cfg to identify your preseed	3
4.2	Writing the preseed file	3
4.2.1	Installing extra packages in your preseed file	4
4.2.2	Installing language support	4
5	Modify pool structure to include more packages	4
6	Create an "Extras" component	4
7	Generating a new ubuntu-keyring .deb to sign your CD	5
8	Building the repository with apt-ftparchive	6
9	Burning the CD	11
9.1	Building the ISO image	11
9.2	Burning the image to CD	11
10	Testing	12
11	Troubleshooting	12

1 Preface

This document is a extract from the officielle documentation of Ubuntu on <https://help.ubuntu.com/community/InstallCDCustomization/>. We estimate that is not interesting to loose time on traducing it, because is understable for us, and our clients don't need read and understand it to be to use our program. We cut from the original document only the part concerning Ubuntu 6.10 and newer. Because we are trying to offer the best and the newest version to our clients.

2 Introduction

The process of customizing or "remastering" Ubuntu install CDs is not especially complex, but it is a little tedious and finicky. This page documents all the problems you might find ; if it doesn't, please edit it !

The Ubuntu install CD (since Ubuntu 6.10, the 'alternative install' or 'server' CD) has three main parts : a bootloader (ISOLINUX on amd64/i386 systems, yaboot on powerpc) and its configuration, which start everything up ; debian-installer (also known as d-i), which in this case is really a specialised miniature Ubuntu system ; and a Debian-style repository structure, which is what takes up all that space on the disk in the directories "pool" and "dists". Building a new CD may involve modifications to all three parts.

3 Copy the CD to your hard drive

Copy the entire installation CD to a directory on your hard drive. This guide will assume your CD image is in `/opt/cd-image/`, but you can put it wherever you like. It will take around 1GB of hard drive space.

If you have an ISO file, you can mount it and copy files out of it without burning it to CD : `mount -o loop /path/to/iso /some/mountpoint`.

You can use **rsync** to copy the CD, or just

```
mkdir -p /opt/cd-image
cp -rT /cdrom /opt/cd-image
```

Be sure to catch the folder `.disk` - if you try and copy `/cdrom/*`, it will ignore the `.disk` folder, as Bash expands the `*` to mean "everything it can see".

4 Modify installer behaviour using a Preseed file

When the CD boots up, a Linux kernel is started and the installation tasks are initiated. The installer's default behaviour can be modified through the use of a "preseed" file, which feeds d-i answers to questions normally asked by debconf, or in other contexts. If you look closely at your install CD, you'll see that certain options (e.g. "server", "expert", "oem") already have preseed files assigned to them.

Suppose you are installing Ubuntu on a bunch of identical computers, and you already know the answers to certain questions (what country and time zone you're in, what keyboard you have, how the network should be configured, how you want to partition the hard disk, etc). You can "preseed" the answers to these questions in a very simple configuration file.

4.1 Changing isolinux.cfg to identify your preseed

We will create a preseed file called 'firewall.seed', which will live in the */preseed* folder of the CD-ROM. We tell d-i where to find this file by modifying the bootloader configuration file, located in *isolinux/isolinux.cfg*, to pass appropriate parameters on the kernel command line. In */opt/cd-image/isolinux/isolinux.cfg*, add a new section labelled like this :

```

LABEL firewall
menu label ^Firewall installation
kernel /install/vmlinuz
append preseed/file=/cdrom/preseed/firewall.seed \
debian-installer/locale=en_NZ kbd-chooser/method=us \
initrd=/install/initrd.gz
ramdisk_size=16384 root=/dev/ram rw quiet --
```

If you want to set the default action to booting with your custom seed, change the DEFAULT line to read 'DEFAULT firewall'.

You must specify a locale and keyboard on the command line, as these questions are asked before the seed is loaded. You can also set DEBCONF_PRIORITY here to ensure you don't see any unnecessary debconf questions.

4.2 Writing the preseed file

A preseed file looks something like this (from the default ubuntu-server.seed on the 6.10 CD)

```

# Always install the server kernel.
d-i base-installer/kernel/override-image string linux-server
# Don't install usplash.
d-i base-installer/kernel/linux/extra-packages-2.6 string
# Desktop system not installed; don't waste time and disk space copying it.
d-i archive-copier/desktop-task string ubuntu-standard
d-i archive-copier/ship-task string
# Only install the standard system and language packs.
d-i pkgsel/install-pattern string ~t^ubuntu-standard$
d-i pkgsel/language-pack-patterns string
# No language support packages.
d-i pkgsel/install-language-support boolean false
```

First notice the format. There are 4 fields :

- identity of the program which will pick up this command
- name of the variable whose value will be passed
- variable type
- value of variable

Warning : d-i expects there to be exactly one tab or space between variable type and variable value (any other space is seen as being part of the value).

The easiest way to create a preseed file is to start with an example and modify it. You could also check out Debian's wiki page on preseeding the installer - 6.10 and higher use the Etch version of the Debian installer.

Note : debconf-get-selections may not be installed on your system. It is part of the debconf-utils package. You may install it thus :

```
sudo apt-get install debconf-utils
```

If you can't find the option you're looking for, try executing the following commands :

```
debconf-get-selections --installer > somefile.txt  
debconf-get-selections >> somefile.txt
```

This will output a list of all debconf options you've chosen throughout your install ; you can pick options out of this and put them into your preseed file.

Warning : debconf-get-selections prints 2 spaces between variable type and variable value. For Breezy, you need to change this to one space before putting the line in a preseed file.

4.2.1 Installing extra packages in your preseed file

base-config has been removed from debian-installer : use

```
d-i pkgsel/install-pattern string ~t^ubuntu-standard$|~n^openssh-server$
```

4.2.2 Installing language support

A different mechanism is used to install additional languages. Preseed the detailed locale question asked by the installer in expert mode. See the first column of */usr/share/i18n/SUPPORTED* for the locale names you can use here. For example, to add support for Bengali and Tamil, use this line :

```
d-i localechooser/supported-locales multiselect bn_IN, ta_IN
```

5 Modify pool structure to include more packages

Probably the prime motivation to build your own install CDs is to modify which packages are installed ; in particular you may want to add some packages to the CD.

The easiest way to do this is to build an 'extras' repository structure, containing only your extra .debs, and merge these into the CD file hierarchy before rebuilding the .ISO image. This guide will step you through how to do this.

This requires you to generate the Packages files that defines what files are in your repository ; the Release file that indexes your Packages files, and the signed Release.gpg file, approving the repository as being official.

6 Create an "Extras" component

Create directories for your new component (substituting your version where appropriate) :

```
cd /opt/cd-image
mkdir -p dists/edgy/extras/binary-i386 pool/extras/ isolinux preseed
```

Put all the extra .debs you want on your CD into pool/extras.

Create the file dists/edgy/extras/binary-i386/Release with the following content :

```
Archive: edgy
Version: 6.10
Component: extras
Origin: Ubuntu
Label: Ubuntu
Architecture: i386
```

On this page <https://help.ubuntu.com/community/InstallCDCustomization/Scripts> there is a useful script that will strip out all of the packages from your CD image that are not currently installed. You will need to run apt-ftparchive (below) to generate the Packages file.

7 Generating a new ubuntu-keyring .deb to sign your CD

In order to sign the Release file, we need to use GPG. The install system will then check the signature against the public keys held in the package ubuntu-keyring. You do not have a private key that matches one of the ones in the shipped ubuntu-keyring, so we need to build a custom version of the ubuntu-keyring package. Install the gnupg package if you do not have it already.

To create a signing key, enter `gpg --gen-key`. Accept the defaults, (for this use, it is probably OK to use "No expiry"). For your Real Name and E-mail address, you might like to use something like "XXX Signing Key" and "**packages@xxx.example.org**". Enter an appropriate passphrase.

In another directory (I use `/opt/build/`), we will download the source for the ubuntu-keyring package, unpack it, add our own GPG key, and rebuild the package. These steps import the 2 Ubuntu public signing keys into your main keyring, then exports them, along with your own public signing key, into a replacement keyring. "YOURKEYID" should be replaced with the 8-digit hexadecimal code that gpg tells you when you do the `--list-keys` command. "Signing Key Name" is what you used in the previous step, when running `gpg --gen-key`.

To clarify, below is an example '`gpg --list-keys`' response. In this example, "YOURKEYID" immediately follows the '/' on the line beginning with 'pub' (which in this example is '437D05B5'.)

```
gpg --list-keys
pub 1024D/437D05B5 2006-09-08
uid XXX Signing Key <packages@xxx.example.org>
sub 2048g/79164387 2006-09-08
```

Note : the following example requires the fakeroot package which may not be installed on your system. You may install it thus :

```
sudo apt-get install fakeroot
```

Here is an example, which you will need to customize to suit your own setup :

```
apt-get source ubuntu-keyring
cd ubuntu-keyring-2005.01.12.1/keyrings
gpg --import < ubuntu-archive-keyring.gpg
gpg --list-keys "Signing Key Name"
gpg --export FBB75451 437D05B5 YOURKEYID > ubuntu-archive-keyring.gpg
cd .. (you are now on ubuntu-keyring-2005.01.12.1)
dpkg-buildpackage -rfakeroot -m"Your Name <your.email@your.host>" -kYOURKEYID
cd .. (you are now on the directory where you started, /opt/build)
cp ubuntu-keyring*deb /opt/cd-image/pool/main/u/ubuntu-keyring
```

You will end up with a udeb file for the installer, and a .deb file for the system. Both files need to be copied into the main component of your CD, because the CD will not check the extras directory.

8 Building the repository with apt-ftparchive

apt-ftparchive builds the Packages and Packages.gz files, needed by the installer. In order to use apt-ftparchive, we will need to provide it with some configuration and some index files.

We will put the index files in */opt/indices* :

```
mkdir -p /opt/indices
cd /opt/indices/
DIST=edgy; wget http://archive.ubuntu.com/ubuntu/indices/override.$DIST.\
{extra.main,main,main.debian-installer,restricted,restricted.debian-installer
```

Create the files "apt-ftparchive-deb.conf", "apt-ftparchive-udeb.conf", "apt-ftparchive-extras.conf", and "release.conf" in a directory (/opt/apt-ftparchive), substituting /opt/cd-image/ for the path to your CD image directory, and /opt/indices/ for the location of the index files, if they differ.

apt-ftparchive-deb.conf

```
Dir {
    ArchiveDir "/opt/cd-image/";
};

TreeDefault {
    Directory "pool/";
};

BinDirectory "pool/main" {
    Packages "dists/edgy/main/binary-i386/Packages";
    BinOverride "/opt/indices/override.edgy.main";
    ExtraOverride "path/to/indices/override.edgy.extra.main";
};

BinDirectory "pool/restricted" {
    Packages "dists/edgy/restricted/binary-i386/Packages";
    BinOverride "/opt/indices/override.edgy.restricted";
};

Default {
    Packages {
        Extensions ".deb";
        Compress ". gzip";
    };
};

Contents {
    Compress "gzip";
};
```

The ExtraOverride component above is needed to add the Task header to main packages, referenced from the preseed file with `t<task name>`.

This is not supplied - but can be extracted from the existing main Packages file with a simple perl script :

extraoverride.pl

```
#!/usr/bin/perl
# extraoverride.pl
# generate ExtraOverride file
# use as follows :-
# extraoverride.pl < /media/cdrom0/dists/edgy/main/binary-i386/Packages > \
# path/to/indices/override.edgy.extra.main

while (<>) {
    chomp;
    next if /^ /;
    if (/^$/ ) {
        print "$package Task $task\n";
    }
    ($key, $value) = split /: /, $_, 2;
    if ($key eq 'Package') {
        $package = $value;
    }
    if ($key eq 'Task') {
        $task = $value;
    }
}
```

apt-ftparchive-udeb.conf

```
Dir {
    ArchiveDir "/opt/cd-image/";
};

TreeDefault {
    Directory "pool/";
};

BinDirectory "pool/main" {
    Packages "dists/edgy/main/debian-installer/binary-i386/Packages";
    BinOverride "/opt/indices/override.edgy.main.debian-installer";
};

BinDirectory "pool/restricted" {
    Packages "dists/edgy/restricted/debian-installer/binary-i386/Packages";
    BinOverride "/opt/indices/override.edgy.restricted.debian-installer";
};

Default {
    Packages {
        Extensions ".udeb";
        Compress ". gzip";
    };
};

Contents {
    Compress "gzip";
};
```

apt-ftparchive-extras.conf

```
Dir {
    ArchiveDir "/opt/cd-image/";
};

TreeDefault {
    Directory "pool/";
};

BinDirectory "pool/extras" {
    Packages "dists/edgy/extras/binary-i386/Packages";
};

Default {
    Packages {
        Extensions ".deb";
        Compress ". gzip";
    };
};

Contents {
    Compress "gzip";
};
```

This is the configuration file for apt-ftparchive. Change to suit your distribution version :

release.conf

```
APT::FTPArchive::Release::Origin "Ubuntu";
APT::FTPArchive::Release::Label "Ubuntu";
APT::FTPArchive::Release::Suite "edgy";
APT::FTPArchive::Release::Version "6.10";
APT::FTPArchive::Release::Codename "edgy";
APT::FTPArchive::Release::Architectures "i386";
APT::FTPArchive::Release::Components "main restricted extras";
```

To build the repository, sign it, and update the MD5SUM file, you can use a script like this :

```
BUILD=/opt/firewall-image-edgy
APTCONF=/opt/apt-ftparchive/release.conf
DISTNAME=edgy

pushd $BUILD
apt-ftparchive -c $APTCONF generate /opt/apt-ftparchive/apt-ftparchive-deb.co
apt-ftparchive -c $APTCONF generate /opt/apt-ftparchive/apt-ftparchive-udeb.c
apt-ftparchive -c $APTCONF generate /opt/apt-ftparchive/apt-ftparchive-extras
apt-ftparchive -c $APTCONF release $BUILD/dists/$DISTNAME > \
$BUILD/dists/$DISTNAME/Release

gpg --default-key "YOURKEYID" --output $BUILD/dists/$DISTNAME/Release.gpg \
-ba $BUILD/dists/$DISTNAME/Release
find . -type f -print0 | xargs -0 md5sum > md5sum.txt
popd
```

9 Burning the CD

At this point, you have a directory which is ready to be collected into an .ISO file and then burnt to a CD.

9.1 Building the ISO image

Under x86 (i386) and x86_64 (AMD64) (for other architecture, please, see the officiel documentation)

```
IMAGE=custom.iso
BUILD=/opt/cd-image/

mkisofs -r -V "Custom Ubuntu Install CD" \
        -cache-inodes \
        -J -l -b isolinux/isolinux.bin \
        -c isolinux/boot.cat -no-emul-boot \
        -boot-load-size 4 -boot-info-table \
        -o $IMAGE $BUILD
```

9.2 Burning the image to CD

To detect the location of your CD drive, try `cdrecord -scanbus`. For a primary IDE CD drive on `/dev/hdc` :

```
sudo nice -18 cdrecord dev=ATA:0,1,0 --speed=24 --blank=fast \
-v -gracetime=2 -tao $IMAGE
```

You will burn a lot of coasters in experimenting with this process, so please use rewritable media !

10 Testing

You can now boot off your CD. Select the new menu item you created. If you’ve used preseeding to stop the CD asking questions, the installation could be totally automatic.

11 Troubleshooting

- If your CD fails to read, try re-burning it, possibly at a slower speed.
- If you get a red d-i screen, check the error message (use Alt-F2, Alt-F3 etc to cycle through the terminals to read the log), and fix your CD appropriately.

ISAIP
18, rue du 8 Mai 1945
49000 St Barthélémy d'Anjou



Dossier Hébergement du projet sur SourceForge

Développement d'une application de personnalisation de système
d'exploitation : "OS A La Carte"

26 Novembre 2006

Mohammed BADISS, Jérôme CHARLOT
2^{ème} année CPI
Promotion 2005 – 2007

Sous la responsabilité de :
M Alexandre LEPRIEULT ;



Table des matières

1 Objectifs	2
2 Présentation	2
3 Modalités d’inscription	2
4 Outils que nous utiliserons	5

1 Objectifs

L'hébergement du projet sur SourceForge permet de donner une lisibilité au projet. SourceForge propose différents outils utiles à la conduite de projets informatiques comme les mailings lists, outils de synchronisation de code entre les différents développeurs, rapport de bogues, ...

2 Présentation

SourceForge regroupe, au mois de Juillet 2006, *130 000* projets et environ *1 350 000* utilisateurs. C'est un hébergeur de programmes Open Source, ainsi pour être accepté dans cette "organisation", il faut développer son projet sous License Open Source.

3 Modalités d'inscription

Il faut d'abord créer un compte développeur sur le site. Puis faire une demande pour le projet. Il faut déterminer les caractéristiques du projet, ainsi nous avons choisis ces options :

– La licence d'utilisation :

License : : OSI-Approved Open Source : : GNU General Public License (GPL)

Note : C'est la licence la plus utilisée dans les logiciels Open Source.

– Le public concerné par le projet :

Intended Audience : : by End-User Class : : Advanced End Users

Intended Audience : : by End-User Class : : End Users/Desktop

– Le statut du développement :

Development Status : : 5 – Production/Stable

– Les types de technologie :

Topic : : Internet : : WWW/HTTP

Topic : : System : : Operating System Kernels : : Linux

– Langage de développement :

Programming Language : : PHP

Programming Language : : Python

– Système d'exploitation concerné :

Operating System : : Modern (Vendor-Supported) Desktop Operating Systems : : Linux

Note : Nous personnaliserons la distribution Ubuntu qui est un système d'exploitation Linux.

– Traduction :

Translations : : French

– Système de gestion de base de données utilisées :

Database Environment : : Network-based DBMS : : MySQL

Par la suite, il est nécessaire de détailler le projet, les technologies que nous utiliserons et une rapide description de l'implémentation que nous comptons faire. Nous avons rédigé cette description :

Project Description

This project, called "OS A La Carte", aims to setup a Linux Operating System from a web page.

On the website, the user can setup administrative tasks, for example defining username and associated password. network settings, default language, and he can also add extra packages.

Once choices have been made, the user has to wait until the end of the ISO modification. Then the user downloads the ISO file, burns it and finally installs the custom Operating System.

The main target of this program are End Users and even Advanced End Users.

This script will modify Ubuntu Linux Distro (the "alternate" flavor).

Implementation

In order to make it possible, we will use PHP interpreted language and Python programming language.

The program will be divided into 2 parts. The first part is the interface, where the user makes his selections, coded in PHP and the second part is a script which the main goal is to custom the install CD using Python (and distribution packages edition).

Cependant un administrateur nous a répondu que notre demande n'était pas assez complète et nous a retourné une dizaine de questions à approfondir. Voici les questions et nos réponses :

We suggest that Sourceforge hosts only the source code (python, php script and SQL schemas) and we will provide the infrastructure. We will deploy Apache web server, Mysql Database with our own dedicated serveur and bandwidth. In order, to coordinate all developpers, we will need subversion with its interface and mailing list.

1. How frequently do you expect to be updating packages within your tree?
Only once, we will sync with Ubuntu release cycle (every 6 months).

2. Do you plan to also provide full releases in an ISO file format (or similar)?
Yes, we will provide it, but sourceforge won't host this ISO file format.

3. How many members of your team will be directly updating the package tree?
Two members will be in charge of updating the package tree.

4. How many distinct versions or quality delineations do you intend to maintain (i.e. stable, unstable, testing, etc.)?
We won't provide distinct version, only one customization flavour.

5. What mechanism will your end-users be using to update their local package load?
End-users will use the same way as they use when they update an original ubuntu distribution. So we don't interfere in ubuntu process, their system will be updated when ubuntu release new updates.

6. How many packages do you expect to exist in each tree (as described in #4)?
We think we will modify about 20 packages (gnome related packages, firefox, ...).

7. What do you estimate to be your maximal space requirement for all materials you intend to provide?
In order to host the source code (php/python scripts), we will need about 50 MB.

8. Are all materials you are providing under an Open Source license?
Every code will be under the GPL license.

9. Does your project intend to make use of our project CVS and/or Subversion services?
It more interesting for us to use subversion services.

10. What other SourceForge.net features will your project be using?
We'd like to use mailing list, bugtracking system and even web space to introduce the project.

Finallement, le projet sera validé par SourceForge le 22 Novembre 2006.

4 Outils que nous utiliserons

Cet hébergement nous permettra d'utiliser un dépôt commun pour mettre le code, ce qui nous permettra de synchroniser le code entre les différents développeurs. Nous utiliserons les outils de rapports de bugs avec les clients et finalement les mailings lists si des utilisateurs nécessitent par la suite des fonctionnalités très spécifiques.

ISAIP
18, rue du 8 Mai 1945
49000 St Barthélémy d'Anjou



The GNU General Public License

Développement d'une application de personnalisation de système
d'exploitation : "OS A La Carte"
10 Décembre 2006

Mohammed BADISS, Jérôme CHARLOT
2^{ème} année CPI
Promotion 2005 – 2007

Sous la responsabilité de :
M Alexandre LEPRIEULT ;



Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law : that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License ; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty ; keep intact all the notices that refer to this License and to the absence of any warranty ; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions :

- (a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- (b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- (c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception : if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you ; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following :
 - (a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange ; or,
 - (b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange ; or,
 - (c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program

subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program. If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims ; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system ; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation ; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE

STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Peut-on encore se passer d'Eclipse ?

Incontournable, cette plate-forme rallie éditeurs et entreprises grâce à l'intégration de ses outils, mais elle impose ses standards "de fait".

Frédéric Bordage , Décision Informatique, le 29/12/2006 à 07h00

Il y a presque cinq ans jour pour jour, IBM créait la fondation Eclipse pour fédérer les développeurs autour d'un seul atelier de développement gratuit et ouvert. Un pari réussi au-delà de ses ambitions : **44 %** des entreprises utilisent aujourd'hui Eclipse IDE (source Forrester). Et, mis à part Microsoft et Sun, tous les éditeurs s'y sont ralliés, tels Borland avec JBuilder et Zend avec le prochain Zend Studio. " Eclipse est déjà un standard dans la communauté PHP ", note Perrick Penet, président de l'Afup (Association française des utilisateurs de PHP). Même l'éditeur spécialiste en gestion de contenu Nuxeo - promoteur de Zope - a intégré le client riche Eclipse RCP à la prochaine version de sa plate-forme, Nuxeo 5.

Pourquoi un tel succès ? Eclipse n'est pas un simple atelier de développement, c'est avant tout une fondation qui réunit les éditeurs les plus puissants autour d'une plate-forme logicielle commune. " Basé sur les spécifications OSGi Service Platform, le cœur d'Eclipse [Equinox, NDLR] permet d'assembler des composants standards pour créer n'importe quel logiciel : un outil de développement tel Eclipse IDE, une application métier comme le fait SAP, etc. ", explique Didier Girard, directeur technique de la SSII Sfeir.

De l'IDE à la plate-forme globale

Comme Linux en son temps, cette approche modulaire a permis de créer très vite un vaste écosystème : les éditeurs et les communautés open source ont transformé leurs outils en plug-in pour Eclipse. Tous adhèrent à la fondation dans l'espoir d'influencer ses projets. Celle-ci compte 152 membres. Elle dispose officiellement d'un budget annuel de 5,7 millions de dollars qui en fait la quatrième communauté open source dans le monde après OSDL (Linux) et les fondations Apache et Mozilla. La fondation Eclipse se consacre à la création d'une plate-forme modulaire couvrant tout le cycle de vie des applications et plus encore. D'abord constituée d'un simple atelier de développement, cette plate-forme regroupe 65 projets, constitués principalement d'outils spécialisés - éditeur de code, test, modélisation, etc. - agrémentés d'un framework.

Les éditeurs de logiciels de développement s'appuient surtout sur Eclipse IDE et ses sous-projets, tandis que les entreprises et les éditeurs de logiciels métier exploitent les frameworks et les socles d'exécution : RCP pour le client riche, RAP pour Ajax, BIRT pour le décisionnel, etc. Ces frameworks accélèrent les développements tout en les standardisant. Cette standardisation progressive intéresse vivement les éditeurs. La fondation Eclipse pourrait ainsi héberger deux nouveaux projets. Le premier, Aperi, vise à simplifier l'administration des environnements de stockage grâce à une interface de programmation (API) standard. Il regroupe de nombreux poids lourds : Brocade, Fujitsu, IBM, Cisco et Net-App. Le second projet, Cosmos, vise aussi à fournir une API standard, mais pour l'instrumentation de n'importe quel service d'une SOA. Son objectif est de faciliter la collecte de données et le développement d'outils d'administration qui partageraient les mêmes structures de données.

Liés à des standards émergents, ces projets comptent sur Eclipse pour dégager rapidement un consensus. Consensus qui prendra la forme d'un framework et d'outils s'imposant progressivement comme des

standards "de fait". La fondation compte sur ces projets pour bâtir un seul entrepôt de données normalisées qui couvrent tout le cycle de vie des applications.

Incontournable... mais problématique

Si elle peut sembler bénéfique, cette standardisation autour d'Eclipse est pourtant dangereuse. En effet, en regroupant les principaux éditeurs de logiciels et en dégagant un consensus restreint, au-delà des instances traditionnelles de normalisation, Eclipse pourrait peu à peu imposer au reste de la planète ses méthodes et ses techniques, qui ne sont pas des standards au vrai sens du terme. Alexis Moussine Pouchkine, architecte Java chez Sun, considère "qu'Eclipse est d'abord une organisation au service de ses membres", et rappelle que le concurrent maison Net-Beans "est toujours le premier outil à implémenter les standards - Java EE 5, EJB 3.0, JAX-WS 2.0, etc. - et uniquement des standards". On peut prendre le cas d'Eclipse RCP (Rich Client Plat-form) qui s'impose souvent pour le développement de clients riches Java. Pourtant, les applications basées sur ce framework ne sont plus portables, car elles imposent le déploiement du framework d'Eclipse RCP sur chaque poste utilisateur.

Les entreprises doivent donc maintenant choisir entre Eclipse RCP et le standard Java SE. Pour elles, il aurait été préférable que les innovations d'Eclipse RCP soient intégrées au sein de Java SE ou dans un framework standardisé par le JCP... En sera-t-il de même pour les autres domaines couverts par les projets de la fondation Eclipse ? La question mérite d'être posée. Car, mis à part Microsoft, aucun éditeur ne semble aujourd'hui en mesure de lutter contre les "standards Eclipse" qui s'imposent progressivement comme des standards "de fait".

Des distributions Eclipse

Le foisonnement de plug-in et l'architecture en couches d'Eclipse ne sont pas faciles à gérer pour les entreprises. Quels plug-in retenir et comment les mettre à jour ? **Face à ces questions, un tiers des entreprises crée et maintient sa propre distribution tandis qu'un autre tiers laisse chaque développeur se débrouiller comme il l'entend.** Le tiers restant attend une version packagée intégrant les plug-in les plus couramment utilisés. Des "distributions Eclipse" commerciales comme celles d'Exadel et de Yoxos apparaissent donc pour combler ce manque. A l'image de Linux, on peut imaginer qu'il en existera demain des centaines, spécialisées dans des domaines précis : téléphonie mobile, client riche, applications métier, BPM, etc.

Source : <http://www.01net.com/article/336966.html>.

B Acronymes

HTML HyperText Markup Language

OVH On Vous Héberge

SGBDR Système de Gestion de Base de Données Relationnelles

API Application Programming Interface

HTTP HyperText Transfer Protocol

FTP File Transfer Protocol

EDI Environnement de Développement Intégré

PHP Hypertext Preprocessor

MCD Modèle Conceptuel de Données

MPD Modèle Physique de Données

SQL Structured Query Language

DVD Digital Versatile Disc

CD Compact Disc

GPL General Public Licence

UML Unified Modeling Language

ISO International Organization for Standardization

CSS Cascading Style Sheet

DHCP Dynamic Host Configuration Protocol

DNS Domain Name System

IP Internet Protocol

BIOS Basic Input Output System

ADSL Asymmetric Digital Subscriber Line

KISS Keep It Simple, Stupide

C Glossaire

Méthode Agile

Une méthode Agile est une méthode de développement informatique permettant de concevoir des logiciels en impliquant au maximum le demandeur (client), ce qui permet une grande réactivité à ses demandes. Les méthodes agiles se veulent plus pragmatiques que les méthodes traditionnelles. Elles visent la satisfaction réelle du besoin du client, et non d’un contrat établi préalablement.

Une distribution linux

Une distribution Linux (ou distribution GNU/Linux) est un ensemble cohérent de logiciels rassemblant un système d’exploitation composé d’un noyau Linux, de logiciels issus du projet GNU et de logiciels supplémentaires - le plus souvent libres.

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.

Une image DVD (au format ISO)

C’est un **format de fichier** qui a pour objectif d’être gravé sur un support matériel par exemple CD/DVD. Comme il s’agit d’un **standard**, l’objectif est d’être portable, il est donc **indépendant** d’un système d’exploitation. Ce qui est important pour notre projet car nous sommes indépendant de la plateforme que le client utilise.

Une machine virtuelle (ou virtualisation)

Dans notre cas d’utilisation, on appelle virtualisation l’ensemble des techniques logicielles qui permettent de faire fonctionner sur une seule machine **plusieurs systèmes d’exploitation** et/ou plusieurs applications, séparément les uns des autres, comme s’ils fonctionnaient sur des machines physiques distinctes. Les outils de virtualisation servent à faire fonctionner ce qu’on appelle communément des serveurs privés virtuels (“Virtual Private Servers”) ou encore environnements virtuels

Une entête “HTTP”

Cette technique fait référence au protocole “HTTP” qui est utilisé pour la **navigation sur Internet**.

Les entêtes font référence aux échanges entre un client et son serveur, typiquement une requête pour consulter un site et la réponse du serveur qui délivre le contenu de la page demandée.

Dans notre projet, les entêtes “HTTP” sont utilisées pour effectuer des **tests**, au niveau du module “SELECT”.

Un paquet

En informatique, et en particulier dans le contexte des systèmes d'exploitation **Unix**, on appelle paquet (ou parfois paquetage, en anglais “package”) une **archive** (fichier compressé) comprenant les fichiers informatiques, les informations et procédures nécessaires à l'installation d'un logiciel sur un système d'exploitation au sein d'un agrégat logiciel, en s'assurant de la **cohérence** fonctionnelle du système ainsi modifié.

Une dépendance (logicielle)

Une dépendance logicielle se conçoit dans le cadre d'une **intégration** de paquet logiciel en vue de construire un agrégat logiciel. Une dépendance exprime des **relations** entre paquets :

- de pré-requis d'inclusion conditionnelle ;
- de post-requis d'inclusion conditionnelle ;
- de hiérarchie d'utilisation (j'utilise, je suis utilisé par) ;
- d'exclusion conditionnelle ;
- de suggestion.

Ceci permet de généraliser une installation logicielle, et de déporter l'intelligence servant à déterminer, et installer les paquets logiciels dans l'ordre, du paquet lui-même vers un outil appelé outil d'installation. Ainsi, cela diminue les efforts des développeurs de paquets logiciels.

Ces dépendances sont un élément nécessaire dans l'architecture de paquets logiciels, qui sont la base des distributions Linux.

Une bibliothèque (logicielle)

En informatique, une bibliothèque logicielle est un ensemble de **fonctions regroupées** pour réaliser un groupe de tâches du même domaine. Les bibliothèques logicielles se distinguent des exécutables dans la mesure où elles ne sont pas complètes, elles ne possèdent pas l'essentiel d'un programme comme une fonction principale et par conséquent qu'elles ne peuvent pas être exécutées directement.

Les bibliothèques peuvent regrouper des fonctions simples (par exemple le calcul d'un cosinus, ou l'inversion d'une matrice) comme des fonctions complexes avec de nombreuses fonctions internes non accessibles directement. Généralement, l'intérêt des bibliothèques réside dans le fait qu'elles peuvent être partagées par plusieurs exécutables.

Les bibliothèques sont parfois appelées **librairies** ; il s'agit d'un abus de langage résultant d'une mauvaise traduction du faux ami en anglais library.

Un dépôt

En informatique, un dépôt (de l'anglais “repository”), est un **stockage centralisé** et organisé de données. Ce peut être une ou plusieurs bases de données où les fichiers sont localisés en vue de leur distribution sur le réseau, ou bien un endroit directement accessible aux utilisateurs.

La plupart des distributions Linux utilisent des dépôts accessibles sur Internet, officiels et non-officiels, permettant aux utilisateurs de télécharger et de **mettre à jour** des logiciels compatibles.

Un bureau

En informatique, on utilise le terme de bureau pour désigner l'espace de travail visible lorsqu'aucune fenêtre n'a été ouverte ou qu'elles ont toutes été réduites, ceci dans le cas d'un environnement graphique. Ce terme a été choisi en comparaison avec la surface de travail disponible sur le meuble portant le même nom.

Généralement, le bureau est la première chose que l'utilisateur peut voir à l'écran une fois qu'il s'est identifié grâce à un écran d'accueil et que l'ordinateur ait fini de démarrer et soit prêt à répondre aux demandes de l'utilisateur.

Un bureau se compose généralement :

- d'un fond d'écran, aussi appelé papier peint (wallpaper en anglais), qui est une couleur, une image ou un motif répété sur la surface du bureau ;
- d'icônes placées directement sur le papier peint ;
- d'une barre accueillant divers éléments (menu, accès aux applications en cours d'exécution, horloge, etc.).

Un bureau virtuel

C'est la possibilité, dans la plupart des environnements graphiques (cela nécessite parfois un logiciel supplémentaire) de gérer plusieurs bureaux, permettant ainsi de mieux gérer l'utilisation de nombreuses applications simultanément.

Ainsi, chaque bureau virtuel est dédié à une action, par exemple la programmation, la navigation sur internet ou autres tâches.

Un cookie

Les cookies sont de petits **fichiers textes** stockés par le **navigateur Web** sur le disque dur du visiteur d'un site Web et qui servent (entre autres) à enregistrer des informations sur le visiteur ou encore sur son parcours dans le site.

Le webmestre peut ainsi reconnaître les habitudes d'un visiteur et personnaliser la présentation de son site pour chaque visiteur ; les cookies permettent alors de garder en mémoire combien d'articles il faut afficher en page d'accueil ou encore de retenir les identifiants de connexion.

Un greffon (ou plugin)

En informatique, le terme anglais plugin (ou plug-in, se prononçant /plɪɡɪn/, du verbe to plug in qui signifie brancher), est employé pour désigner un programme qui interagit avec un logiciel principal, appelé programme hôte, pour lui apporter de nouvelles fonctionnalités.

Le terme plugin étant un anglicisme, de nombreux termes français ont été proposés pour le remplacer : module externe, module enfichable, module d'extension, extension, greffon ou encore plugiciel (terme proposé par l'Office de la langue française du Québec en 1996).

Une session

En informatique, une session est l'exécution d'un programme pour un utilisateur donné. L'exécution du programme est alors paramétrée par les informations du profil de l'utilisateur (ses caractéristiques, ses préférences, l'historique de ses interactions avec le programme, etc.)

Une liste de diffusion (ou mailing list)

Une liste de diffusion ou liste de discussion, terme parfois abrégé en lidi(e) (mailing list en anglais, abrégé en ML) est une utilisation spécifique du **courrier électronique** qui permet le publipostage (en masse) d'informations à un grand nombre d'utilisateurs possédant une adresse courriel.

Un logiciel, le serveur, est installé sur un ordinateur qui traite les courriels entrants (qu'il reçoit), et, selon leur contenu, soit effectue des actions spécifiques, soit (cas général) distribue le message à toutes les personnes abonnées à cette liste de diffusion.

Un serveur graphique

Il s'agit de la “couche” la plus élevée dans le système d'exploitation, qui s'occupe notamment de l'affichage.

Le serveur graphique est un système qui gère l'interaction **homme-machine** par l'écran, la souris et le clavier de certains ordinateurs en réseau. C'est généralement un standard ouvert d'interaction graphique avec l'utilisateur sur les Unix, Linux et les *BSD.

BIOS

Le “Basic Input Output System” ou BIOS (système élémentaire d'entrée/sortie) est un programme contenu dans la mémoire morte (ROM) de la carte mère s'exécutant au démarrage de l'ordinateur. Il s'occupe des composants de l'ordinateur comme les disques, et autres périphériques. Sa tâche principale est de fournir un support de bas niveau pour communiquer avec les périphériques.

ADSL

ADSL signifie “Asymmetric Digital Subscriber Line” en anglais, mais la définition française est “Ligne d'abonné numérique à débit asymétrique”.

Il s'agit de la technologie la plus répandue en France pour les connexions internet haut débit.

EDI

Un environnement de développement intégré (EDI ou IDE en anglais pour Integrated Development Environment) est un programme regroupant un éditeur de texte, un compilateur, des outils automatiques de fabrication, et souvent un débogueur.

Bien que des EDI pour plusieurs langages existent, bien souvent (surtout dans les produits commerciaux) un EDI est dédié à un seul langage de programmation.

On peut également trouver dans un EDI un système de gestion de versions et différents outils pour faciliter la création de l’interface graphique (GUI en anglais pour Graphical User Interface).

D Bibliographie

Références

- [1] SourceForge. Plate-forme de développement de projets libres.
<http://www.sourceforge.net>.
- [2] DELL. Vendeur informatique qui propose de personnaliser la configuration matérielle de la machine.
<http://www.dell.com/>.
- [3] System76. Vendeur informatique de portables avec la distribution linux ubuntu, Octobre 2006.
<http://www.system76.com/>.
- [4] Christian Rolland. *L^AT_EX par la pratique*. O'Reilly, 1999.
Livre permettant de concevoir l'ensemble de ce rapport, L^AT_EX étant un langage de composition de documents.
- [5] Kyle Rankin et Bill Childers Jonathan OXER. *Ubuntu Hacks : Tips & Tools for Exploring, Using, And Tuning Linux*. O'Reilly, Juillet 2006.
Livre permettant de configurer le serveur linux et de sécuriser celui-ci face aux attaques extérieures.
- [6] Michel Casabianca. *Extreme Programming (précis & concis)*. O'Reilly, 2005.
Livre de support pour notre méthode de conduite de projet “Agile”.
- [7] Laurent Bourdon Andrew Ford. *Apache (précis & concis)*. O'Reilly, 2000.
Livre de support pour la configuration du serveur Web Apache.
- [8] James Guérin Rasmus Lerdorf. *PHP (précis & concis)*. O'Reilly, Mars 2003.
Livre d'aide pour les bases du langage PHP.
- [9] Giles Carré Chuck Toporek. *Python (précis & concis)*. O'Reilly, 2002.
Livre référentiel pour le langage de développement Python.
- [10] Warren W. Gay. *Programmation Linux langage C*. CampusPress, 2004.
Livre référentiel pour l'utilisation des outils linux (makefiles).
- [11] Tarek Ziadé. *Programmation Python, Syntaxe, Conception et optimisation*. Eyrolles, 2006.
Livre de support pour la conception des différents modules en Python.
- [12] Eric Daspet Cyril Pierre de Geyer. *PHP 5 avancé*. Eyrolles, 2006.
Livre de support pour la conception de l'interface Web (module “Select”).
- [13] Ubuntu/Canonical. Custom install cd, tous les scripts de base qui permettent de personnaliser le cd d'installation.
<https://wiki.ubuntu.com/InstallCDCustomizationHowTo>.
- [14] Ubuntu/Canonical. Site officiel de la distribution linux ubuntu.
<http://www.ubuntu.com/>.
- [15] Frédéric Bordage. Peut-on encore se passer d'eclipse ?, 29 Décembre 2006.
<http://www.01net.com/article/336966.html>.
- [16] Free. Offre free dédibox.
<http://www.dedibox.fr/>.
- [17] OVH (On vous héberge). Offre ovh “kimsufi”.
<http://www.kimsufi.com/>.
- [18] Gandi. Offre d'un an pour un nom de domaine (15 €.).
<http://gandi.net/>.

- [19] Distrowatch. Parts de marché des différentes distributions linux.
<http://www.distrowatch.com>.
- [20] Mandriva. Support proposé par mandriva.
<http://www.mandriva.com/en/linux/2007/powerpack>.
- [21] Mandriva. Prix de la distribution mandriva édition powerpack.
http://store.mandriva.com/product_info.php?products_id=341.
- [22] Novell. Prix de la distribution mandriva édition powerpack.
http://www.novell.com/products/desktop/pricing_euro.html.
- [23] Alsacréations. Alsacréations est une communauté dédiée à la conception web aux normes, et notamment aux standards *W3C*, aux feuilles de style *CSS*, aux langages *HTML* et *XHTML*, ainsi qu'à l'accessibilité du *Web* en général.
<http://www.alsacreations.com/>.
- [24] OpenWeb. Informations sur les principaux standards du *Web* et les navigateurs qui les respectent. article sur les popup.
<http://www.openweb.eu.org/articles/popup/>.
- [25] Equipe Grappa Université de Lille. Faq $\text{\LaTeX}$ de l'équipe grappa, 2001.
<http://www.grappa.univ-lille3.fr/FAQ-LaTeX/>.
- [26] Wikipedia L'encyclopédie libre. Définition d'une dépendance logicielle, 2007.
http://fr.wikipedia.org/wiki/D%C3%A9pendance_logicielle.
- [27] Wikipedia L'encyclopédie libre. Définition d'une distribution linux, 2007.
http://fr.wikipedia.org/wiki/Distribution_linux.
- [28] Wikipedia L'encyclopédie libre. Définition du concept de la virtualisation, 2007.
<http://fr.wikipedia.org/wiki/Virtualisation>.
- [29] Wikipedia L'encyclopédie libre. Définition d'un paquet, 2007.
http://fr.wikipedia.org/wiki/Paquet_%28logiciel%29.
- [30] Wikipedia L'encyclopédie libre. Définition d'un dépôt, 2007.
http://fr.wikipedia.org/wiki/D%C3%A9p%C3%B4t_%28informatique%29.
- [31] Wikipedia L'encyclopédie libre. Définition d'une bibliothèque, 2007.
http://fr.wikipedia.org/wiki/Biblioth%C3%A8ques_logicielles.
- [32] Kurt Cagle. Utilisation des attributs visibility & display, Janvier 2000.
<http://www.devx.com/tips/Tip/13638>.
- [33] Ubuntu. exemple de fichier de préconfiguration "preseed", Octobre 2006.
<https://help.ubuntu.com/6.10/ubuntu/installation-guide/i386/preseed-contents.html>.
- [34] UbuntuForum. Configuration du logiciel de virtualisation (k)qemu, Décembre 2006.
<http://doc.ubuntu-fr.org/qemu?s=kqemu>.
- [35] Equipe PHP. Utilisation de la fonction popen en php, Novembre 2006.
<http://fr2.php.net/manual/en/function.popen.php>.