

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 :

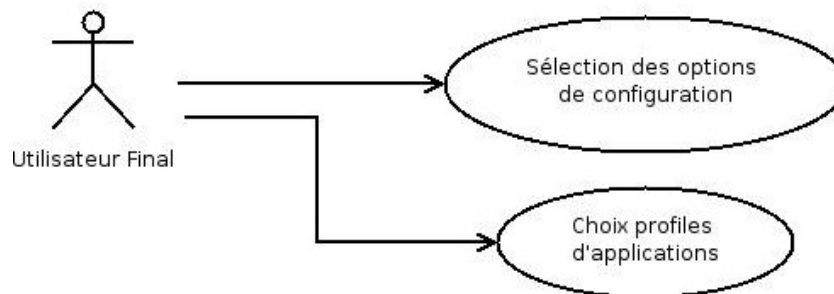


FIG. 1 – Diagramme UML – Cas d'utilisation du projet OS A La Carte.

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.

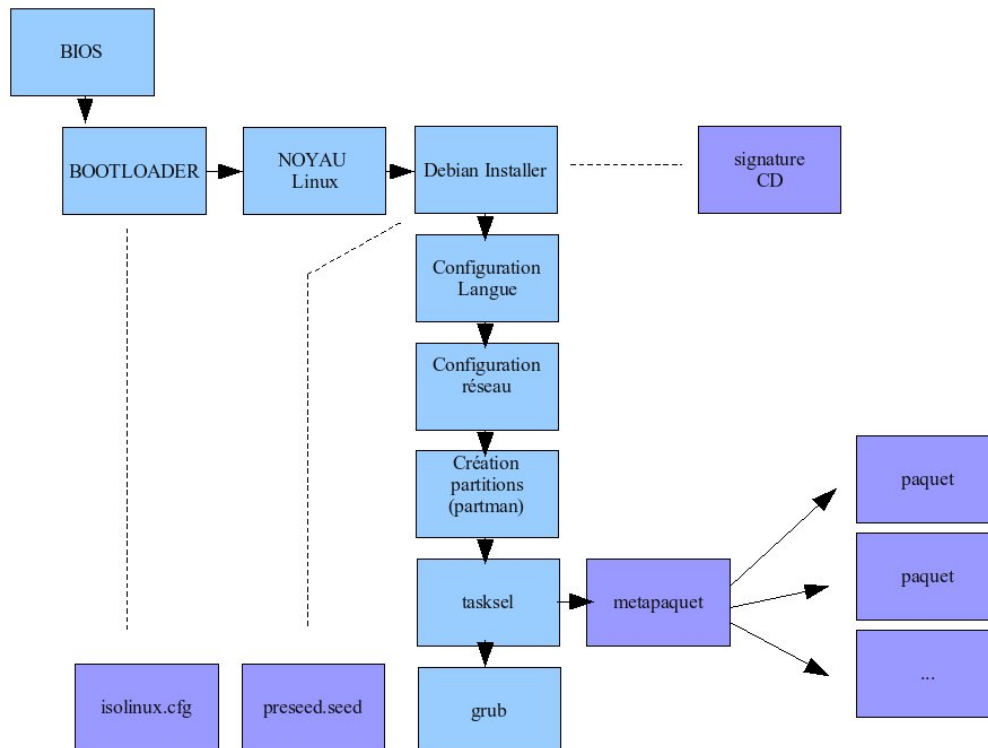


FIG. 2 – Schéma – Différentes étapes du procédé d'installation.

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 **misés à jour de sécurité**. Notre projet qui devra vraisemblablement sortir entre deux versions finales, au plutard en février 2007, devra intégrer les **misés 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 :

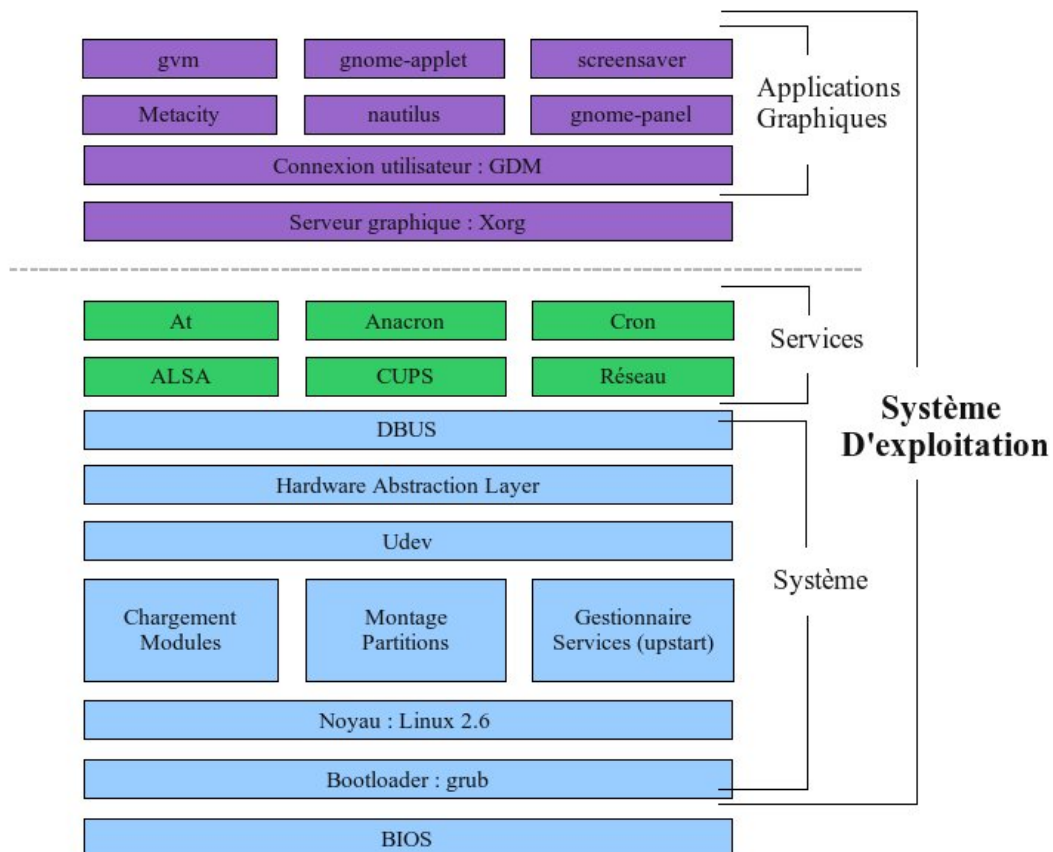


FIG. 5 – Schéma – Domaine d'intervention du projet OS A La Carte.

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 :

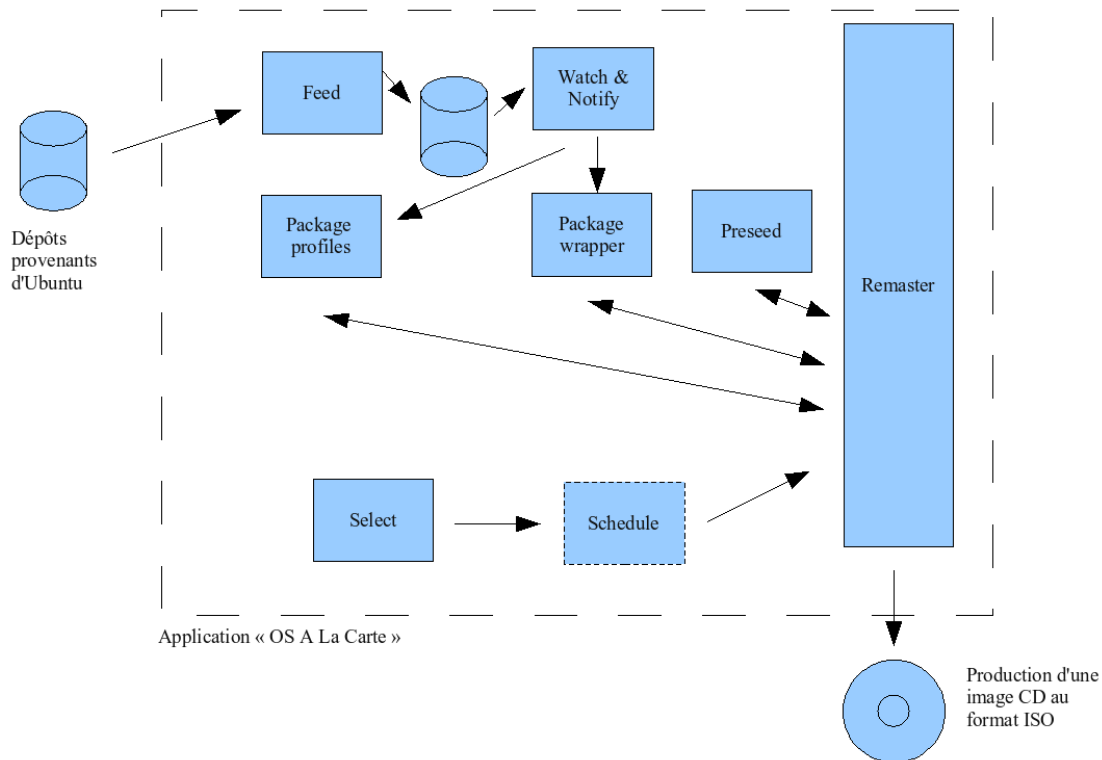


FIG. 6 – Schéma – Découpage du projet OS A La Carte en modules.

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](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.