

Rapport de stage



Développement d'assistants de configuration de serveurs et du système
Mandrake Linux

Arnaud Desmons
Epitech 1 - Promo 2006

1^{er} juillet au 31 décembre

Table des matières

1	Résumé	3
2	Présentation de l'entreprise :	
	MandrakeSoft	5
2.1	Secteur d'activité	5
2.1.1	Linux	5
2.1.2	Les premières distributions	6
2.2	MandrakeSoft	7
2.2.1	Historique	7
2.2.2	MandrakeSoft dans le monde	8
2.2.3	MandrakeSoft en chiffres	8
2.2.4	MandrakeSoft et copyright	8
2.3	Positionnement du stage dans les travaux de l'entreprise	9
3	Travail effectué :	
	Développement d'assistants de configuration de serveurs et du système Mandrake Linux	11
3.1	<i>Drakwizard</i>	11
3.1.1	Modules existants actuellement	11
3.1.2	Cahier des charges	12
3.1.2.1	Modules Perl (.pm)	12
3.1.2.2	Fichiers de description d'interface (.wiz)	12
3.1.2.3	Mode standard et expert	13
3.1.2.4	Internationalisation	13
3.1.3	Statut	14
3.1.3.1	Éléments conservés	14
3.1.3.2	Éléments modifiés	14
3.1.3.3	Éléments supprimés	14
3.1.3.4	Éléments ajoutés	14
3.1.3.5	Restant à faire	15
3.1.4	Conclusion	15
3.2	<i>Drakconnect</i>	16
3.2.1	Cahier des charges	16
3.2.1.1	But général	16
3.2.1.2	Objectifs	16
3.2.2	Compte-rendu	16
3.2.2.1	Résultats	16
3.2.2.2	Conclusion	17

3.3	<i>OpenRouter</i> et Naat v3	18
3.3.1	But général	18
3.3.1.1	OpenRouter	18
3.3.1.2	SNF, MNF et Firewall 3	18
3.3.2	Cahier des charges	18
3.3.2.1	Backend	18
3.3.2.2	Frontend :	18
3.3.3	Objectifs	19
3.3.4	Résultats	19
3.3.5	Statut	20
3.3.5.1	Éléments ajoutés	20
3.3.6	Conclusion	20
3.4	Packaging	21
3.4.1	RPM	21
3.5	Projets externes	22
4	Conclusion	23
4.1	Conclusion technique	23
4.2	Conclusion générale	23
5	Bibliographie	26
6	Glossaire	27

Chapitre 1

Résumé

MandrakeSoft est une société informatique éditant le système d'exploitation Mandrake Linux. Ce dernier comprend le noyau Linux ainsi qu'une série d'applications tout d'abord aux sources disponibles librement.

Au cours de ce stage de six mois, du 1^{er} Juillet au 31 Décembre, chez MandrakeSoft, j'ai été successivement en charge de trois projets principaux sur le thème de la configuration du système Mandrake Linux, plus particulièrement des serveurs et plus généralement du système d'exploitation Linux.

La configuration et en particulier celle des serveurs sous Linux est parfois fastidieuse et le nombre et la complexité des paramètres déroutent parfois les utilisateurs même les plus avertis. Il s'agissait d'améliorer la convivialité de la distribution Mandrake Linux grâce à une palette d'outils de configuration simple pour l'utilisateur final et évolutifs et modulaires pour les développeurs. Voici donc la description de ces projets et la contribution que j'y ai apportée :

Le premier projet qui m'a été confié est DrakWizard. C'est un outil de configuration de serveurs. Le programme en lui-même ne configure rien mais permet le chargement et l'utilisation de modules spécifiques. Il existe aujourd'hui une dizaine de modules configurant chacun un type de service du système. Une précédente version était programmée en langage Java mais n'était plus mise à jour depuis plusieurs années pour de multiples raisons. Ce projet consistait à poursuivre la traduction du code en langage Perl. Une version déjà en partie traduite du programme Drakwizard existait avec environ 30% des fonctionnalités requises pour l'ensemble des modules. J'ai donc finalisé cette traduction du programme en lui-même mais aussi de celle des modules, ce qui représentait le plus gros du travail. Cette nouvelle version permettra une mise à jour plus facile.

J'ai ensuite intégré l'équipe en charge du projet Naat. C'est un outil d'administration de machines en local ou sur les réseaux via, pour l'instant, une interface Web. Il s'agit d'une refonte totale par rapport aux deux précédentes versions. En vue de son intégration dans divers projets il devait faire les preuves de son exploitabilité. Après finalisation du moteur et implémentation de quelques modules il a pu être présenté et validé par une commission. Au sein de l'équipe de quatre personnes, j'étais en charge de cette finalisation et de la création des modules.

Enfin et pour clôturer mon stage, j'ai travaillé sur DrakConnect. C'est un outil de configuration des connexions réseaux pour le système Mandrake Linux (ADSL, modem, câble, LAN) utilisé dès l'installation. Afin de permettre une meilleure intégration de cet outil au sein de la distribution, les informations de configuration du système sont désormais extraites du système lui-même. Auparavant, lors de la mise à jour des paramètres, un fichier propre à *Drakconnect* contenait un doublon des informations afin de faciliter la récupération des informations lors d'une prochaine exécution du logiciel. Il a donc fallu réaliser les parseurs nécessaires à la lecture de la quasi-totalité des paramètres réseaux de la distribution (en fonction des logiciels de connexion proposés dans celle-ci),

directement à partir du système. Ensuite, le remplacement de l'ancien parseur par les nouveaux a été effectuée.

En parallèle de ces projets, j'étais en charge d'un certain nombre de logiciels inclus dans la distribution Mandrake Linux. Les dernières évolutions de ces derniers (nouvelles versions, correctifs, optimisations) sont intégrées dans la distribution grâce à un format spécifique de packaging. Il est alors très facile pour l'utilisateur d'ajouter, de supprimer ou de mettre à jour un logiciel de la distribution.

Ce stage, bien qu'axé sur un aspect particulier de la distribution, à savoir la création d'assistants de configuration, m'a permis de découvrir diverses activités et modes de travail de Mandrake-Soft. D'autre part, les rouages de la configuration du système Linux sont complexes et intéressants et permettent de mieux comprendre les Unix en général.

Chapitre 2

Présentation de l'entreprise : *MandrakeSoft*

2.1 Secteur d'activité

MandrakeSoft se définit à la fois comme un animateur de projets et un fédérateur de compétences dans le domaine de l'Open Source. Son ambition est de faire bénéficier les utilisateurs des innovations issues des travaux de la communauté, qui réunit plusieurs centaines de personnes à travers le monde. En proposant un système d'exploitation puissant Mandrake-Linux garantit une installation et une « prise en main » simples, rapides et fiables.

Linux est le seul système d'exploitation Unix dont la part de marché augmente chaque année. Ses utilisateurs se répartissent en différentes catégories :

1. La première catégorie se compose d'utilisateurs inconditionnels qui sont, en règle générale, des connaisseurs de l'informatique, capables d'apprécier la stabilité et les performances du système à sa juste valeur.
2. De plus en plus de néophytes s'intéressent à une solution complète (système d'exploitation et applications logicielles) qui leur assure la performance, la stabilité, la simplicité et la convivialité à un coût très abordable. L'essor et la multiplication de supports presse généraliste dédiée à Linux attestent de l'intérêt du grand public pour Linux.
3. Les PME/PMI : outre l'aspect économique non négligeable du logiciel libre (frais de licences réduits), Linux fonctionne sur n'importe quelle plate-forme, même techniquement dépassée.

La possibilité d'accéder au code source du produit par la GPL fait de Linux le choix N°1 pour les logiciels embarqués ou intégrés à tout type de produit (PABX, moniteur, boîtier TV, téléphone mobile, organisateur personnel...).

Les principaux concurrents de MandrakeSoft sont les distributions Redhat et SuSe qui à elles trois représentent 80% du marché.

2.1.1 Linux

En 1991, un étudiant finlandais, Linus Torvald, de l'université d'Helsinki achète un PC équipé d'un processeur i386. Ce dernier diffère des précédents processeurs de la gamme d'Intel par un nouveau mode de fonctionnement, le mode protégé. Dans ce mode, le processeur vérifie chaque accès mémoire et interdit aux applications d'adresser des zones mémoire qui ne leur ont pas été allouées, d'où une sécurité et une stabilité du système accrues.

De plus, il brise enfin la limite de la mémoire adressable par les programmes, bridée à 1 Mo depuis le i8086 (conçu en 1979) en autorisant les programmes à adresser la mémoire sur 32 bits soit 4 Go de mémoire.

Malheureusement, le système le plus répandu à l'époque, MS-DOS (et sa sur-couche graphique Windows) est incapable d'utiliser ce mode et fonctionne toujours en 16bits, d'où une vitesse et une stabilité insatisfaisantes.

Linus Torvald voulu utiliser les possibilités de ce nouveau processeur. Attiré par la stabilité et la puissance des systèmes Unix, malheureusement rendus inaccessibles par leur prix (une licence monoposte de Solaris coûtait alors la bagatelle de 10 000 FF), L. Torvald se tourna vers un clone d'Unix à bas prix, Minix.

Cependant, Minix était optimisé pour fonctionner sur de très petites configurations (8086 équipés de 512ko de mémoire). Bien que plus stable et plus rapide que MS-DOS, il restait incapable d'utiliser les possibilités avancées du 386. Or, bien que téléchargeable avec son code source gratuitement sur Internet, Minix n'était pas modifiable et son auteur n'acceptait que rarement les patches fournis par les utilisateurs.

Aussi Linus Torvald se tourna-t-il vers la réalisation de son propre système, Linux. Si les premières annonces étaient assez ironiques (par exemple en faisant appels aux nostalgiques de l'époque où « *les vrais hommes concevaient eux-mêmes leur pilotes de périphériques* ») et si à l'époque (août 1991), Linux n'était qu'un petit hobby, il attira peu à peu de nombreux programmeurs de talent alléchés par l'ouverture du projet (quiconque pouvant contribuer du code), si bien qu'au bout de six mois, Linux était devenu un système aussi complet que Minix mais plus rapide.

Linux était lancé sur ses rails, et désormais rien ne pouvait plus l'arrêter.

Du fait de son modèle de développement ouvert (n'importe qui peut contribuer au programme), Linux s'est enrichi de nombreuses fonctionnalités au fil des années : temps-réel, systèmes répartis, systèmes embarqués, support de nombreuses architectures (680x00, ix86, Alpha, Sparc, ARM, HP-PA, ...), support SMP (multi-processeurs), ...

2.1.2 Les premières distributions

A l'origine, chaque utilisateur devait télécharger les sources du noyau, des bibliothèques et des outils qu'il désirait avant de les compiler depuis un autre système pour pouvoir obtenir un système utilisable. Outre que cela imposait de posséder un autre système d'exploitation et surtout une suite de développement capable de produire du code pour un autre système, tous les utilisateurs ne pouvaient pas se permettre de télécharger plusieurs mégaoctets de programmes.

De plus, cette méthode convenait aux développeurs du système mais pas aux utilisateurs finaux, qui mois après mois, devenaient de plus en plus nombreux, au fur et à mesure que la disponibilité d'un système stable, rapide, puissant, peu gourmand en ressources matérielles (CPU, mémoire et disques) et de surcroît gratuit se répandait.

Aussi les premières distributions de Linux apparurent. Parmi celles-ci, deux sont célèbres et existent toujours, Slackware et Ydrassil. Le distributeur se charge de compiler tous les programmes nécessaires, les archive et fournit un programme d'installation (à l'époque rudimentaire).

Ces distributions sont librement téléchargeables depuis Internet ou peuvent être commandés auprès de revendeurs, celui-ci l'ayant téléchargé gratuitement ou l'ayant acheté auprès du distributeurs qui se rémunère sur la vente de disquettes ou de CDROMs.

Les premières distributions de Linux se présentaient sous la forme d'une ou deux dizaine d'archives contenant chacune plusieurs programmes : Système de base, X Window, Outils de développement, Bases de données, Gestion des imprimantes, ...

Le principal apport de Redhat à Linux consiste en un système de packages appelé RPM (Redhat Package Manager : Système de gestion de programme de Redhat). Ses caractéristiques sont

telles que ce format a été adopté par nombre de distributions (Redhat, Suse, Mandrake, ExecutiveLinux, ...).

Une archive RPM n'est relative qu'à un programme. Il s'agit d'une archive contenant le ou les programmes, la documentation et les fichiers de configuration associés. Lorsqu'une mise à jour s'impose, il n'est plus nécessaire de recréer une énorme archive mais seulement celle relative à ce programme. Le second apport du format RPM qui pallie cette fragmentation des programmes à installer (et donc la complexité de l'installation que cela induit) consiste à inclure dans cette archive, le nom, la version, une description de ce programme. Afin de faciliter la mise à jour des programmes, chaque archive RPM contient également le nom et l'adresse email de celui qui l'a créé, la date de compilation ainsi que la machine ayant servi à compiler le programme.

Redhat fut ainsi la première distribution à créer une base de donnée des programmes installés. Lorsqu'un programme doit être installé, le programme rpm vérifie que celui-ci n'est pas déjà installé et si oui, vérifie que la version installée est bien plus ancienne que celle que l'administrateur veut installer. De plus, rpm assure la gestion des conflits entre programmes : il est interdit de désinstaller une bibliothèque dont des programmes ont encore besoin ou d'installer des programmes dont certains fichiers ont déjà été fournis par d'autres packages.

De ce fait, le format RPM permet une installation plus flexible de Linux et une administration simplifiée des programmes disponible.

En 1998, trois étudiants décident d'améliorer la distribution RedHat en lui adjoignant KDE, une surcouche de X Window plus conviviale, dont l'interface utilisateur est moins étrangère aux utilisateurs de MacOS ou Windows.

La première version de la distribution Mandrake, la 5.1, était née ¹.

2.2 MandrakeSoft

2.2.1 Historique

Au cours de l'été 1998, trois étudiants, Frédéric Bastok, Gaël Duval et Jacques Lemarois, créent une nouvelle distribution de Linux, Mandrake, durant leur temps libre. Au contraire des distributions existantes de Linux, Mandrake vise le vaste marché du grand public et non seulement le segment des serveurs professionnels.

L'accent est mis en particulier sur la convivialité de l'interface homme-machine. Après l'installation, l'utilisateur se retrouve directement sous X-Window, l'interface graphique de Linux, sans avoir à taper de fastidieuses lignes de commandes. L'interface de X-Window fournie par défaut est KDE, un environnement graphique moins austère que ce qui est habituellement fourni sous Linux.

La Mandrake 5.1 (car basée sur une RedHat 5.1) était sortie.

Devant l'engouement suscité par ce concept, les trois créateurs de Mandrake fondent une société en novembre 1998, MandrakeSoft. Linux Mandrake 5.2 fit alors son apparition sur le marché à la fin de l'année 1998, suivi par la version 5.3 en février 1999.

Au printemps 1999, MandrakeSoft a fusionné avec BeroLinux (distribution allemande) pour mettre en commun leurs points forts, convivialité de la Mandrake et optimisations de la BeroLinux.

En juin 1999, une nouvelle version de la distribution la 6.0 souligne l'orientation professionnelle de Linux avec l'intégration en standard du support des disques RAID, un nouveau noyau 2.2 optimisé pour les machines multi-processeurs, une optimisation de tous les programmes pour les Pentiums, ...

Depuis, MandrakeSoft développe une distribution toujours plus conviviale :

- installation graphique (*DrakeX*),

¹(voir page n°7 pour l'historique de MandrakeSoft)

- programme de partitionnement graphique permettant de libérer (de manière non destructive) de la place sur un disque entièrement occupé par Windows pour Linux (*DiskDrake*),
- programme de configuration graphique du matériel (*Harddrake*),
- programme de configuration de la carte graphique permettant de configurer de manière optimale l'écran automatiquement (*XFdrake*),
- logiciels pré-configurés,
- installation de Linux sur une partition Windows (donc plus de problème de partitionnage) (*Linux4Win*)
- facilité de mise à jour (*MandrakeUpdate*)

Ces logiciels sont intégrés dans la distribution au fil des versions : 6.0 (*Harddrake*, *Linux4Win* bêta), 6.1 (*DrakeX*, *DiskDrake*, *XFdrake*, *Linux4Win* final).

A l'automne 1999, une nouvelle distribution, Linux Mandrake Gold 2000, est lancée en destination des PME (avec des logiciels commerciaux en supplément).

Simultanément, la distribution Mandrake de Linux, qui n'existait que pour l'architecture ix86 d'Intel voit son portage pour Alpha (DEC/Compaq) et PowerPC (IBM/Motorola) débiter. Le portage de la Mandrake sur Alpha est déjà très avancé. La dernière version de Mandrake Linux en date est la 9.0.

2.2.2 MandrakeSoft dans le monde

MandrakeSoft est présente dans plusieurs pays :

- Allemagne (depuis la fusion avec BeroLinux),
- Canada
- Chine
- France
- USA (Sillicon Valley)
- Chine via un partenariat avec King Star et LinuxOne dans le but d'améliorer la localisation de Linux en Asie (affichage de polices spécialisées et gestion des caractères chinois et japonais).

2.2.3 MandrakeSoft en chiffres

MandrakeSoft comprend 80 salariés dont 25 % de développeurs. En particulier, MandrakeSoft rémunère des programmeurs travaillant sur de grands projets Open Source (et qui bénéficient donc à tous les utilisateurs de Linux) comme le noyau de Linux, KDE ou KOffice.

Son capital est de 6 000 000 euros. MandrakeSoft est cotée en bourse.

Le chiffre d'affaire de MandrakeSoft au cours de son premier exercice fiscal est de 4MF.

2.2.4 MandrakeSoft et copyright

MandrakeSoft vend plusieurs produits :

- MANDRAKE GPL, ne comprenant que des logiciels libres sous licence GPL ou du même type (BSD, X-MIT, ...)
- POWERPACK MANDRAKE, qui comprend la Mandrake GPL, un CD d'applications supplémentaires (les *contribs*, sous licence GPL), deux CDs contenant les sources des applications précédentes et deux CDs d'applications commerciales (à usage limité dans le temps ou démonstrations)
- PROSUITE ÉDITION, qui comprend 7 CDs et un DVD : la Mandrake GPL, ses sources, les deux CDs de contribs et un CD d'applications commerciales à usage professionnel.

La majorité des applications fournies par Mandrake sont des logiciels libres, à commencer par le coeur du système, le noyau Linux. On appelle logiciels libres, par opposition aux applications propriétaires (ou commerciales), tout logiciel sous licence GPL (ou l'une des licences qui lui sont semblables comme celle des outils BSD ou celle du MIT pour X-Window, X-MIT).

Les licences classiques protègent les droits de ses auteurs. La licence GPL protège, elle, les droits des utilisateurs, à savoir :

- le droit de copier tout logiciel GPL,
- le droit de redistribuer (gratuitement ou non) tout logiciel GPL
- le droit de posséder les sources du logiciel et de les modifier,
- le droit de redistribuer officiellement ces modifications.

De ce fait, tous les logiciels libres que distribue MandrakeSoft sont librement téléchargeables sur son site Web (et ses miroirs). Les applications commerciales ne peuvent être obtenues que sur CD, excepté certaines applications au statut intermédiaire comme Netscape.

MandrakeSoft est une société à but lucratif, son intérêt est donc d'avoir des recettes plus importantes que ses dépenses. Mais MandrakeSoft a également pour but, comme toutes les distributions de Linux (RedHat, Slackware, Debian, Epoch, TurboLinux, ExecutiveLinux, Suse, ...), de soutenir le mouvement des logiciels libres.

Aussi Mandrake rémunère t-elle plusieurs des développeurs des grands projets Open Source comme le noyau Linux, la suite bureautique KOffice, ...

De plus, MandrakeSoft contribue aux logiciels libres en général. A chaque fois que les programmeurs de Mandrake modifient l'un des programmes pour un problème de sécurité, corriger une bogue ou apporter de nouvelles possibilités au logiciel, ils communiquent leurs modifications aux auteurs des-dits logiciels afin que tous les utilisateurs puissent bénéficier de ce travail.

Enfin, MandrakeSoft a lancé de grands projets pour faciliter l'installation et l'usage de Linux : DrakeX, DiskDrake, Harddrake, XFdrake, MCC, MandrakeUpdate, ...

2.3 Positionnement du stage dans les travaux de l'entreprise

Ce stage de développement portait sur les outils de configuration propres à Mandrake. J'ai intégré dans un premier temps, une équipe de sept développeurs (Warly, Tv, Pixel, Francois Pons, Gc, Daouda, Till), directement responsables de ses outils : les DrakXTools. Pour le projet Naat c'est l'équipe sécurité, recherche et développement comprenant quatre personnes qui m'a accueilli.

MandrakeSoft à pour objectif de rendre sa distribution Mandrake Linux accessible au plus grand nombre, de cela découlent plusieurs règles pour ses outils :

1. Localisation : La distribution Mandrake Linux est traduite en plus de 50 langues et à 100% pour le programme d'installation. MandrakeSoft participe à cette traduction en utilisant les spécifications d'internationalisation i18n de G.N.O.M.E.
2. Fiabilité : Un département y est consacré : la QA. Tout les développeurs Mandrake travaillent sur (dans tout les sens du termes ;-) une version de la distribution contenant les dernières mises à jour, appelée Cooker, ainsi ils s'échangent de nombreux retours de bogues. De plus, plusieurs semaines avant la sortie d'une nouvelle version, la distribution est "gelée", c'est à dire qu'on ne peut ajouter de nouvelles fonctionnalités aux outils Mandrake et mettre à jour les packages que pour des corrections de bogues.
3. Convivialité : Mandrake Linux fut la première distribution basée sur KDE. Ce dernier offre une interface utilisateur relativement proche de celle de Windows. De plus, la majorité des outils Mandrake utilisent des bibliothèques communes, ainsi l'utilisateur utilise une interface uniforme ce qui permet une meilleure compréhension et une prise en main plus rapide de ces outils.

4. Facilité de maintenance : La majorité des outils Mandrake sont développés en Perl, c'est un langage qui a l'avantage d'être concis et qui ne nécessite pas de compilation d'où un gain de temps non négligeable pour le portage de la distribution ou lors de la mise à jour de certaines librairies². De plus il se prête très bien aux tâches d'administration Unix et Linux en s'inspirant de nombreuses fonctionnalités de ces systèmes telle que les expressions régulières et la gestion des fichiers, c'est donc un langage de premier choix pour des outils de configuration. Enfin et pour les raisons évoquées précédemment Mandrake dispose d'une importante ressource de développeur Perl. En effet le programme d'installation de la distribution, le gestionnaire de packages Urpmi ainsi que le centre de contrôle Mandrake sont tous réalisés en Perl ainsi que de nombreux outils Mandrake accessibles via le centre de contrôle.

Les outils réalisés lors de ce stage s'inscrivent directement dans cette optique.

²La compilation de la totalité de la distribution et du noyau Linux prend déjà plusieurs jours.

Chapitre 3

Travail effectué : Développement d'assistants de configuration de serveurs et du système Mandrake Linux

3.1 *Drakwizard*

La première version de *DrakWizard* (Wizdrake) était basée sur un projet de 1998 externe à Mandrake (Ggui). Cette première version fut abandonnée car elle était programmée en Java et qu'il n'y avait donc plus personne pour la maintenir. Une version Perl a été initiée par Mael Dodin, ex-stagiaire, j'ai donc succédé directement à son travail en suivant ses directives d'évolution. Cette version comprenait la lecture des fichiers de description des interfaces (fichier XML déjà existants) avec un début de navigation et quelques widgets et les scripts shell de la version java. Ces derniers chargés d'agir au niveau le plus bas pour la lecture et écriture de la configuration du système ont été conservés temporairement. Il n'existait qu'une version en anglais.

3.1.1 Modules existants actuellement

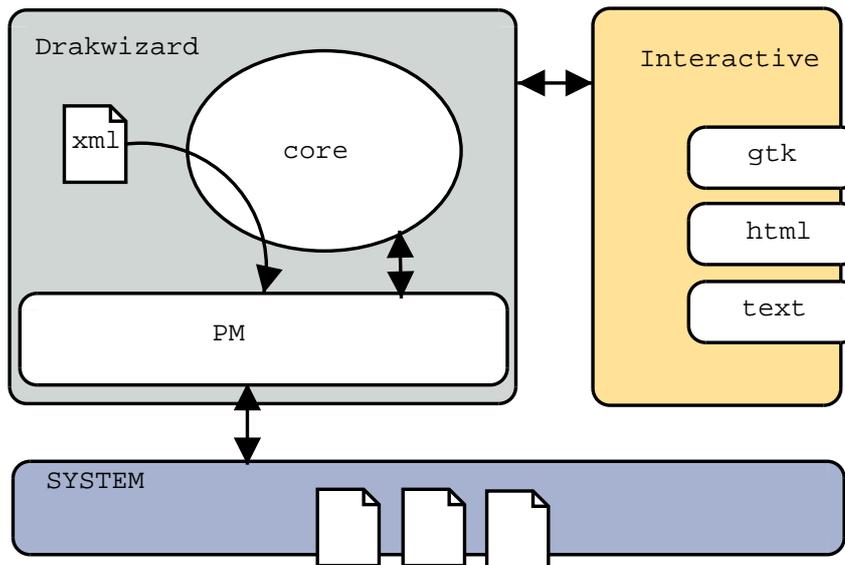
1. Samba : partage de répertoires et d'imprimantes, restriction d'accès, configuration du groupe de travail, partage des répertoires des utilisateurs pour eux-même.
2. FTP : restrictions d'accès, choix d'un répertoire public à partager, partage des répertoires des utilisateurs pour eux-même.
3. HTTP (serveur web apache) : choix de la source des documents, de l'activation et du choix des répertoires utilisateurs.
4. NTP (serveur de temps) : choix de la zone de temps et du serveur de synchronisation.
5. Leafnode (serveur de news) : pseudo cron¹ pour le rapatriement en local de news (fréquence et serveur).
6. Postfix : choix du serveur smtp distant et d'alias mail.
7. DHCP : choix de la plage d'IP à attribuer et nom du domaine.
8. DNS : serveur : choix des serveurs vers lesquels rediriger les requêtes, nom du domaine, client : ajout d'une IP associée à un nom.

¹Permet de lancer des commandes différées.

9. Firewall : Basé sur d'anciens scripts de configuration du firewall Linux (iptables) : Bastille. Bastille n'est désormais plus utilisé dans la distribution le module est donc inutilisable par défaut. Il faut au préalable installer Bastille. De plus, il existe d'autres outils de configuration pour le script plus récent inclus actuellement dans la distribution : Shorewall.

3.1.2 Cahier des charges

L'idée novatrice du projet est qu'il peut être adapté à différents assistants de configuration. En effet la description des différentes pages est passée dans des fichiers XML. Pour obtenir un nouvel assistant, il suffit donc de créer un nouveau fichier XML contenant la description de l'interface (navigation, éléments graphiques, texte, questions, packages nécessaires, etc...) et un module Perl avec les fonctions nécessaires à la modification ou à la lecture de la configuration du système.



3.1.2.1 Modules Perl (.pm)

Ils contiennent les fonctions à la fois pour récupérer des informations venant du système et pour modifier la configuration du système en fonction des informations fournies par l'utilisateur via l'interface. Ces fonctions peuvent être appelées via le fichier de description d'interface.

3.1.2.2 Fichiers de description d'interface (.wiz)

Les fichiers de description d'interface décrivent au format XML l'agencement des éléments graphique à l'intérieur des différentes pages et la navigation entre celles-ci. Des fonctions du module Perl peuvent être appelées tantôt pour les éléments graphiques (remplissage de champs par des valeurs par défaut, listes dynamiques, éléments grisés, etc...) tantôt pour la navigation (enchaînement des pages en fonction des choix de l'utilisateur ou de la valeur de retour de la fonction).

Navigation Le principal problème du projet réside dans la navigation. Dans les fichiers XML toutes les pages décrites ne doivent pas forcément être affichées, certaines sont affichées dans des cas particuliers.

Il y a plusieurs solutions pour afficher la page suivante :

- La page suivante n’est autre que la page qui vient après dans le fichier XML, dans ce cas on affiche page+1.
- La page courante possède une variable “JumpPage” qui contient le nom de la page qui doit être affichée après.
- La page courante appelle une fonction du module Perl, qui selon sa valeur de retour, indique le nom de la page qui doit être affichée. Le nom de la page est alors associé à la valeur de retour via la balise “Target” (le nombre “jumpIndex” est associé au nom de page “targetName” à l’intérieur de cette balise).

Éléments graphiques (widgets)

- Case à cocher : décrit avec la balise “Boolean” dans le fichier de description d’interface. Il renvoie vrai ou faux au module Perl.
- Listes (“Chooser”). Permet de choisir parmi une liste de choix. Elle peut être dynamique, c’est à dire qu’elle est composée d’éléments retournés par une fonction du module perl.
- Champ de text (“Input”) : Permet la saisie et la récupération de chaînes de caractères
- Texte simple (“Output”) : Concerne tout le texte non-éditable.

3.1.2.3 Mode standard et expert

Le mode standard est un mode pas à pas avec plusieurs pages et une aide pour chacune d’elles. Selon les choix de l’utilisateur certaines pages seront ignorées.

Le mode expert se lance en ajoutant l’option –summary en ligne de commande et regroupe toutes les pages en une seule, l’aide n’apparaît plus mais en laissant la souris sur un widget on peut voir apparaître une aide si elle a été renseignée dans la balise XML de description du widget en question. Selon les choix de l’utilisateur certains widgets seront grisés.

D’un point de vue technique, à propos du mode expert, plusieurs solutions ont été testées. Dans un premier temps, un robot effectuait toute la navigation (comme en mode standard) pour déceler les informations inutiles en fonctions des différents choix de l’utilisateur et ainsi griser ces dernières. Or cette solution s’est avérée beaucoup trop lente et consommatrice de mémoire système. Il a donc fallu ajouter des informations d’inter-dépendance entre les éléments graphiques dans le fichier XML de description des interfaces graphiques de manière à griser les uns en fonctions des valeurs des autres.

3.1.2.4 Internationalisation

Un système d’extraction des chaînes de caractères contenues dans les modules et leur centralisation dans un fichier spécifique de base de données permet sa traduction dans toutes les langues y compris à base de caractères spéciaux. Le processus de traduction est commun avec d’autres outils Mandrake ou non. Il est basé sur les spécifications i18n² du projet GNOME. Dans la base, la chaîne de caractères en anglais sert de clé. Une convention dans la définition des chaînes de caractères à l’intérieur du programme permet leur traduction. *Drakwizard* étant modulaire, les chaînes sont contenues dans les fichiers de description d’interface. J’ai donc créée un robot pour extraire ces chaînes et permettre ainsi leur traduction selon le schéma classique défini par i18n. Le robot n’entre donc pas dans ce schéma.

²i18n correspond à l’abréviation du mots “internationalisation” d’un point de vue phonétique.

3.1.3 Statut

Ayant hérité d'un projet en cours de développement et évolutif voici un état des lieux de ce qui existait, de ce qui à été effectué au cours de ce stage et quelques perspectives d'évolution.

3.1.3.1 Éléments conservés

- L'interface graphique (en utilisant la bibliothèque interactive).
- Le moteur de parsing des XML a été très largement enrichi, en même temps que le XML lui même, pour toutes les fonctionnalités ajoutées.
- La base reste identique.
- L'utilisation des lib (lib interactive, XMLparser).

3.1.3.2 Éléments modifiés

- La navigation entre les différentes pages du wizard (débugue)
- Le XML : Principalement : l'appel des fonctions du module Perl via une nouvelle balise ("func").

3.1.3.3 Éléments supprimés

- Les informations relatives à la présentation dans le XML (police, alignement, taille). En effet, le but est de garder une certaine cohérence entre les wizards. Il est préférable de nommer les paramètres à appliquer (à la manière des titres sous Word, ou du CSS) pour pouvoir les modifier en bloc.

3.1.3.4 Éléments ajoutés

Selon les directives du stagiaire me précédant les éléments suivants ont été ajoutés. Pour chacun d'entre eux, un extrait de la documentation réalisée par Mael Dodin au terme de son stage y explicite les raisons de ses suggestions.

- Traduction en Perl

“Pour toute la navigation et la configuration le programme utilise une multitude de scripts shell et cela implique de nombreux appels système. Le langage Perl permet de faire exactement la même chose que les scripts shell, il y a donc peu d'intérêt à les utiliser, il est plus judicieux de rester en Perl et de n'utiliser qu'un seul langage plutôt que de “jongler” constamment entre les deux.” Mael Dodin

- Simplification du XML (moulinette)

“Ces fichiers ont été écrits à l'origine en fonction d'une interface Java. Il ne sont donc pas totalement adaptés à une interface Perl. Beaucoup de paramètres sont inutiles. Toutes les indications d'ergonomie deviennent donc inutiles. Toutes ces améliorations permettraient de simplifier le code Perl, car la récupération des informations contenues dans les fichiers XML est parfois compliquée.” Mael Dodin

- Installation des packages (cf balise “rpm”)

“L'utilisateur doit avoir installé les différents packages avant de pouvoir configurer les serveurs, or il ne sait pas forcément quels packages devraient être installés. Il faudrait donc gérer les dépendances pour que les packages soit automatiquement installés avant le lancement de la configuration. L'utilisateur n'aurait donc plus à se préoccuper du choix des packages.” Mael Dodin

- Les listes déroulantes standard
- l’appel récursif de wizards
- les listes dynamiques de cases à cocher
- Les listes déroulantes dynamiques en Perl
- Le grisement dynamique des widgets en fonctions des autres
- Le mode expert.
- L’internationalisation.

3.1.3.5 Restant à faire

- Utiliser la libconf pour structurer davantage le projet.
- Intégration au projet Naat en uniformisant le XML.

3.1.4 Conclusion

Drakwizard dépend largement de la puissance des parseurs (cf libconf et autres projets). On peut envisager de proposer à l'utilisateur grâce à *DrakWizard* tout ce que les parseurs permettront à la fois en matière de lecture de la configuration du système actuellement ainsi que de sa mise à jour.

Différences entre les parseurs actuels de *Drakwizard* et ceux d'autres projets : Les parseurs actuels s'apparentent à un ensemble de fonctions capable d'activer ou de désactiver une fonctionnalité du serveur ou de changer des paramètres bien définis. Or généralement les parseurs transforment un fichier de configuration et un seul bien précis sous forme de structure interne à un langage. Il reste donc au programme à "comprendre" l'ensemble des ces données et leurs interactions pour pouvoir modifier sans la compromettre une configuration déjà existante.

Adoption par la communauté : *Drakwizard* est désormais inclu dans la distribution Mandrake Linux avec actuellement une dizaine de modules. De plus, des développeurs travaillent actuellement bénévolement à la réalisation de nouveaux modules (LDAP par exemple).



mode standard



mode expert

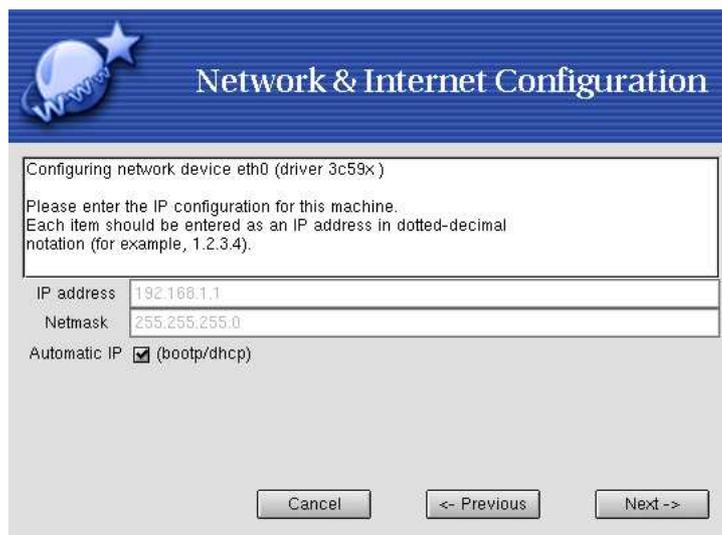
3.2 *Drakconnect*

3.2.1 Cahier des charges

3.2.1.1 But général

Drakconnect est un assistant de configuration des connexions réseaux pour le système Mandrake Linux. Il est utilisé dès l'installation du système, c'est donc un outil majeur de la distribution.

Il permet la configuration des connexions modem (PPP, ADSL, câble) et LAN (Ethernet). Il propose une aide pas à pas pour l'utilisateur. Certains champs sont déjà remplis et correspondent à la configuration actuelle du système.



3.2.1.2 Objectifs

En complément de la modification du système propre à la configuration choisie *Drakconnect* maintenant à un fichier utilisable par d'autres programmes, propres à la distribution, pour connaître les paramètres réseaux sélectionnés par l'utilisateur. Il s'agissait de développer les fonctions nécessaires au rapatriement des informations de configuration des connexions, initialement contenues dans ce fichier, directement à partir du système. Ceci pour tenir compte d'éventuelles modifications de la configuration réseau effectuées hors de *DrakConnect*. Ces fonctions sont contenues dans des bibliothèques et pourront donc être utilisées dans les programmes qui se basaient sur le fichier de configuration de *Drakconnect*. Cela permet une plus grande souplesse et indépendance des outils au sein de la distribution car il n'est plus requis de faire appel exclusivement à *Drakconnect* et donc de l'installer pour utiliser les autres outils Mandrake. De plus, les modifications effectuées en dehors du cadre de *Drakconnect*, c'est à dire manuellement ou par des programmes tiers, pourront être intégrées par ce dernier et plus généralement par les outils utilisant ces nouvelles bibliothèques, c'est à dire la majorité des outils Mandrake.

3.2.2 Compte-rendu

3.2.2.1 Résultats

Près de 90% des informations auparavant contenues dans le fichier de configuration de *Drakconnect* sont maintenant obtenues directement à partir du système grâce aux nouveaux parseurs.

Les 10% restant sont à diviser en deux parties : L'une, incompressible, comprend des informations propres au programme lui-même. Une autre partie pourra diminuer quant à elle, par l'ajout de nouvelles fonctionnalités. En effet, pour la première sous catégorie, des informations telles que la connexion actuellement par défaut sont indispensables car impossibles à déduire du système : il existe par exemple plusieurs moyens disponibles pour l'utilisateur d'activer une connexion, on ne peut donc pas déduire la connexion qui sera activée. Cette partie est donc dépendante des possibilités du système d'exploitation. La seconde partie en revanche pourra disparaître au fil des évolutions du logiciel car elle est dépendante de ses fonctionnalités. Actuellement, le nom d'utilisateur de la connexion est conservé dans ce fichier alors que le mot de passe est déduit, on pourrait envisager de supprimer cette information en laissant le choix entre tous les identifiants pour une connexion donnée.

3.2.2.2 Conclusion

La quantité de code ajouté ou supprimé ne se compte qu'en dizaines de lignes. La difficulté résidait surtout dans l'importance et la complexité du code à analyser. En effet, la modification du système était déjà existante, seule la lecture ne se faisait qu'à partir d'un fichier (mis à jour en même temps que le système). Il fallait donc s'inspirer du code utilisé pour mettre à jour le système pour faire l'opération inverse et ainsi obtenir des informations à partir du système. La partie écriture a également été simplifiée par la suppression de ce fichier désormais inutile. La complexité du code réside dans le fait que de nombreuses couches de bibliothèques interviennent et que les modifications à apporter se trouvaient à un niveau relativement bas du système.

3.3 *OpenRouter* et Naat v3

3.3.1 But général

Naat (Network Appliance Administration Tool) a pour but de créer un firewall basé sur la distribution Mandrake Linux administrable via différents types d'interfaces (Web, texte, graphique). Actuellement seule l'interface Web est prise en charge mais son architecture permet une conversion facile vers tout type de format de description d'interface.

3.3.1.1 OpenRouter

Dans le cadre d'un appel d'offre Européen (IST : Information Society Technologies) Mandrake fut chargé de la partie Software du projet OpenRouter. C'est une machine destinée aux petits et moyens réseaux servant de passerelle vers Internet, elle prend également en charge un certain nombre de tâches serveur comme le DHCP ou le cache DNS. Il s'agit de fournir la plate-forme logicielle, c'est à dire Linux et les différents logiciels (DHCP, DNS, etc), issus du logiciel libre ainsi que l'interface Web qui permettra une configuration rapide et simple du router. La partie logicielle dépend de contraintes matérielles inhérentes à ce type de machine "légère" (Espace disque et mémoire limité, architecture ARM). L'interface de configuration choisie fut Naat 3.

3.3.1.2 SNF, MNF et Firewall 3

Firewall 3, nommé ainsi provisoirement est la future version de firewall basé sur Mandrake Linux, elle succédera ainsi à la SNF (Single Network Firewall) et la MNF (Multi Network Firewall) sortie très récemment. La première ne gère que deux réseaux (Internet et Intranet), la seconde accepte, en plus, une DMZ. Cette dernière évolution est permise par l'intégration au sein de la distribution d'un nouveau script de configuration de firewall Linux (Iptables) gérant plus de trois interfaces : Shorewall.

3.3.2 Cahier des charges

Le projet se découpe en deux parties principales : un "backend" qui écrit et lit les informations de configuration de la machine à paramétrer et un "frontend", la partie interface. Chaque partie peut être exécutée sur une machine différente.

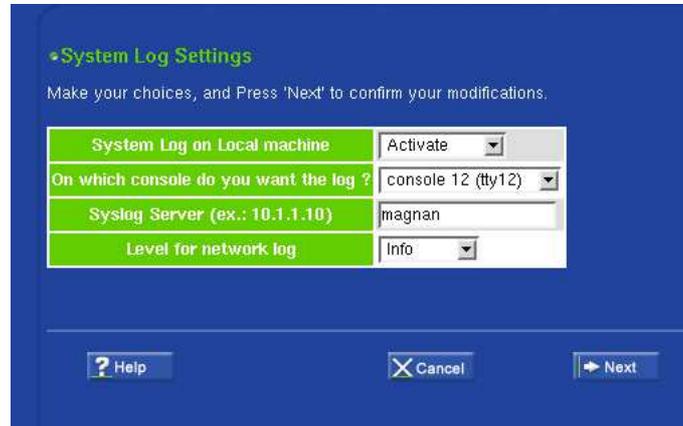
3.3.2.1 Backend

C'est un serveur qui répond au format XML aux requêtes du même format provenant du frontend pour consulter ou modifier la configuration du système. Il comprend des modules chargés de lire directement la configuration du système pour la convertir en XML et de modifier le système en fonction des modifications de l'arbre XML.

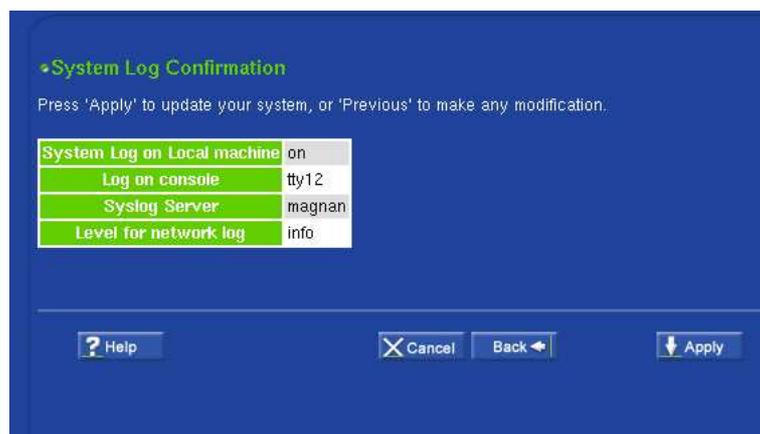
3.3.2.2 Frontend :

Il gère la transformation du XML provenant du backend en différents formats pour l'interface (HTML notamment) via le langage de Transformation XSL (Extensible Stylesheet Language) : XSLT. Seul l'HTML est géré pour le moment. Il gère également l'interaction avec l'utilisateur comme la navigation entre et dans les assistants de configuration. Il comprend des modules de description des interfaces de configuration. Il y a une complémentarité et une dépendance entre la partie backend d'un module et sa partie frontend. Il s'agit de réaliser les modules à la fois dans

le backend et le frontend qui permettront de réaliser la configuration des différents services (firewall, DHCP, DNS pour OpenRouter). XML permet une représentation standard de l'information (comme de l'interface ou de la configuration du système ou de la gestion des transactions et de la navigation) alors que XSL permet sa traduction dans d'autres formats (HTML, libGlade, etc...). Une fois l'information traduite en XML elle peut être manipulée plus facilement.



System Log Settings	
System Log on Local machine	Activate
On which console do you want the log ?	console 12 (tty12)
Syslog Server (ex.: 10.1.1.10)	magnan
Level for network log	Info



System Log Confirmation	
System Log on Local machine	on
Log on console	tty12
Syslog Server	magnan
Level for network log	info

3.3.3 Objectifs

Philippe Hetroy avait réécrit en totalité la partie backend, le frontend ne nécessitant pour sa part que des modifications. La partie backend et frontend était à finaliser et les modules de configuration à implémenter. En vue de la présentation du projet à une commission européenne, il s'agissait de démontrer les capacités de la plate-forme Naat en bouclant la chaîne des transactions allant du frontend au backend avec en exemple quelques modules.

3.3.4 Résultats

L'architecture de Naat étant très modulaire, l'apprentissage des différentes technologies s'est faite par étape et selon la nécessité. Dans un premier temps, les parseurs issus en partie du projet Drakwizart ont été intégrés au backend. Ensuite les modules du frontend ont été portés de la version précédente vers Naat. Au fil des bogues, l'étude du moteur a permis leur résolution et enfin le développement de nouvelles fonctionnalités telles que les tableaux dynamiques ou l'utilisation de variables dans le XML.

On peut maintenant considérer le moteur validé, reste donc à développer davantage de modules pour rendre le projet OpenRouter entièrement administrable et exploitable.

3.3.5 Statut

La précédente version est sortie récemment sur la distribution Mandrake MNF (www.mandrakesecure.net) et succède directement à la première version de Naat sur la distribution SNF (Single Network Firewall). J'ai pour ma part travaillé exclusivement sur la version 3.

Ce projet a été réalisé au sein d'une équipe composée de quatre personnes notamment en charge des précédentes versions de Naat.

Cette dernière version avait été commencée depuis quatre mois. Un stagiaire me précédant avait réécrit toute la partie backend, le frontend ne nécessitant pour sa part que des modifications.

3.3.5.1 Éléments ajoutés

- Le module de configuration DHCP
- Le module de configuration du serveur DNS
- Le module de réglage de l'horloge système.
- Le module de paramétrage des interfaces réseaux
- Le module de configuration du serveur de logs.
- La gestion des tableaux dynamique en XML/XSLT.

et au fur et à mesure de l'implémentation des modules : débogue du Framework et nettoyage du code (traduction en objet).

3.3.6 Conclusion

La prise de connaissance de ce projet complexe de par son architecture était fastidieuse pour plusieurs raisons : Aucune documentation n'avait été rédigée contrairement à la version précédente ; De plus, beaucoup des technologies employées m'étaient encore inconnues (XSL, XPATH et LibXML, organisation des modules pour le backend et le frontend) ; Enfin la mauvaise gestion des informations de débogue ne permettait pas de "tracer" correctement le déroulement, bon ou mauvais, du programme.

3.4 Packaging

En parallele de mes projets et comme tout les developpeurs Mandrake, j'etais responsable d'un certains nombre de package. J'ai aussi mis a jour ou initié quelques packages qui ne m'avait pas été assignés, par interet, par curiosite informatique ou par prestige ;-).

- slrn-pull : Offline news reading support for slrn
- xsoldier : Shooting game on X Window System
- icecast : Streaming Media Server
- tuxpuck : Clone of ShufflePuck Cafe historical game
- drakwizard : Wizard Launcher and its collection of wizards
- firewall3 : Network Appliances Administration Tool
- eMusic : Sound player for the Enlightenment window-manager
- emacs-htmlize : HTML-ize font-lock buffers
- xboard : An X Window System graphical chessboard
- xpenguins : Cute little penguins that walk along the tops of your windows
- gtkhx : GtkHx is a GTK+ version of Hx, a UNIX Hotline Client
- fwbuilder : Firewall administration tool
- foobillard : OpenGL billard game
- gtkgraph : Simple GTK graphing calculator
- pixieplus : An image and photo viewer, browser, manager and simple editor
- as6edriver : Linux driver for the Artec AS6E parallel port interface scanner
- wmwebcam : Watch your own webcam in a small dockapp window
- swish-e : Simple Web Indexing System for Humans - Enhanced
- slrn : A powerful, easy to use, threaded Internet news reader
- emacs-tnt : Emacs based AOL Instant Message Client
- pinball : 3d Pinball game and its level editor
- nessus : Nessus security scanner

3.4.1 RPM

Les packages sont des fichiers rpm qu'il s'agit de generer à partir des sources distribuées sur Internet. Il en existe 2 formes : On distingue les RPM, RedHat Package Manager des SRPMS ou Sources RPM. Les RPM simples ne necessitent aucune compilation prealable. Il contient la forme binaire du logiciel. Il ne peuvent etre distribués seuls selon la GPL, il y a donc le SRPM, qui permet apres compilation l'obtention du RPM. L'un des avantages majeur du RPM est qu'il permet une simple distribution, administration des programmes. Un fichier, le spec file determine comment construire le RPM a partir des sources et comment repartir dans l'arboresence du systeme de fichier les binaires generés ainsi que les fichiers de configuration. Il contient aussi le nom, la description, la version et bien d'autres choses encore concernant tout ce qu'il est utile de savoir pour l'administration du logiciel en question. On peut alors supprimer ou mettre a jour "proprement" (la base rpm connaissant precisement à quel package appartient tel fichier) sansrisque de conflit (grace a la balise Conflicts du spec file) et en gerant les dependances entre les packages (ex on a besoin de Gtk pour installer une application utilisant cette interface) grace a la balise Require et Build-Require (qui concerne pour sa part les dependances a la compilation).

3.5 Projets externes

Les projets internes précédemment évoqués sont tous liés à la configuration du système Linux et par conséquent fortement basés sur des techniques de parsing. En conséquence, il m'a fallu essayer, étudier et parfois même contribuer à d'autres projets en relation avec ces techniques.

Config4Gnu <http://config4gnu.sourceforge.net>

C'est un concurrent direct de DrakWizard. Il fournit les mêmes fonctionnalités mais avec davantage de couches d'abstraction ce qui permet son utilisation avec LibConf. Déjà plusieurs types d'interfaces sont disponibles (Web, ligne de commande, graphique).

Libconf <http://project.tuxfamily.org/libconf>

C'est une API écrite en Perl qui fournit un niveau d'abstraction supplémentaire par rapport aux fichiers de configuration.

Webmin <http://www.webmin.com>

C'est une interface d'administration de systèmes Unix via le Web qui à l'avantage de proposer de très nombreux modules.

E-smith http://www.mitel.com/products/product.cfm?p_id=76

En moins de dix minutes SME transforme un PC compatible intel en un serveur réseau tournant sous Linux pour les petites et moyennes entreprises. Ses fonctionnalités incluent : firewall, partage de fichier et d'imprimante, email et Web sécurise et administration distante via une interface Web.

GladeXML <http://developer.gnome.org/doc/API/libglade/gladexml.html>

Permet de charger des interfaces dynamiques décrites en XML.

Chapitre 4

Conclusion

4.1 Conclusion technique

Le système Linux, et les Unix en général, en dépit de leur relative souplesse, sont parfois, du moins sur certains points, trop complexes. Les assistants, en rendant plus accessible ce paramétrage, le rendent cependant plus obscure aux yeux de l'utilisateur et plus limité. Il s'agit donc d'ajuster le niveau de complexité par défaut à l'utilisateur visé tout en gardant une transparence suffisante pour que tous y trouvent leur compte. La politique adoptée généralement par MandrakeSoft, y compris pour le projet Drakwizard, est la présence d'un mode standard et expert pour la majorité de ses outils. D'autres solutions, comme la génération d'un fichier de configuration personnalisé à combiner, grâce aux informations contenues en commentaires avec le système, permettent à l'utilisateur une meilleure appréhension des modifications qu'il effectue. Au-delà de la complexité pour un type de paramétrage donné, il n'existe aucun standard de configuration à la manière de la base de registre sous Windows. Certains projets se destinent à une distribution unique dans le but d'uniformiser la configuration des outils de cette distribution (Libconf, Config4Gnu et Gentoo). Il est utile de sensibiliser les programmeurs à l'importance d'un standard car les éditeurs de distribution seules ne peuvent effectuer ce travail pour tous les logiciels qu'elles proposent. Il est certain qu'une distribution permettant cette configuration centralisée susciterait l'intérêt du plus grand nombre et c'est actuellement le cas de la MNF et de la distribution E-smith pour les routeurs et les serveurs. Cependant certaines technologies favorisent cette uniformisation dans la représentation de l'information et par conséquent des informations de configuration. C'est par exemple le cas du XML et des interfaces d'administration Web en HTML. On peut ainsi passer de l'un à l'autre via d'autres technologies de transformation comme le XSLT. C'est très exactement ce que réalise la projet Naat.

4.2 Conclusion générale

Ce stage ne m'a permis que très faiblement d'appliquer mes compétences techniques spécifiques mais de les élargir très largement. L'enseignement Epitech m'a été très bénéfique ici, où j'ai été confronté à une situation semblable à celle des premiers mois de piscine où l'on doit tout apprendre : se documenter, formuler des questions précises, etc...

Ce stage peut se résumer en une somme d'enrichissements techniques, culturels, personnels et humains. De plus, bien que positive, cette expérience m'a permis de noter quelques critiques qui, quoique négatives, seront, je l'espère, plus constructives pour la suite de mes études, futurs stages et à plus long terme dans ma vie professionnelle .

En effet, d'abord d'un point de vue technique, c'est en autodidacte que j'ai acquis des compétences nouvelles dans les domaines du système Linux et de sa configuration, de la programmation, des outils de développement Open Source et de la gestion de projet. La plupart des outils de configuration Mandrake étant réalisés en Perl j'ai dû en faire l'apprentissage. Plus généralement et en raison de la forte dépendance entre les outils de configuration Mandrake et donc de leurs bibliothèques, la programmation était orientée objet. Il m'a aussi fallu comprendre et intégrer une quantité importante de code. Ma première mission, dans le cadre du projet Drakwizard était de traduire en Perl des scripts shells, j'ai donc approfondi mes connaissances dans ces deux langages. Les scripts shell faisant appel à d'autres programmes de parsing (sed, grep, etc...) j'ai dû comprendre leur fonctionnement en vue de leur remplacement par des scripts Perl. Lors du projet OpenRouter, l'utilisation du XML était prépondérante et m'a permis d'acquérir des compétences dans ce langage et ses dérivés (XSLT, XPATH). J'étais également en charge d'un certain nombre de logiciels. J'ai donc appris à packager au format RPM différents logiciels qui m'étaient attribués. Pour chacun d'eux il m'a fallu suivre leurs évolutions (nouvelles versions, correctifs, etc...). Cela m'a permis de découvrir des logiciels que je ne connaissais pas.

Culturellement, le fait d'utiliser des produits Open Source dans mes projets (bibliothèques, langages) et pour mes projets (CVS, RPM, Linux) m'a fait découvrir des méthodes de travail disponibles librement et efficaces.

D'autre part, humainement, le travail en équipe, quoique rare, m'a permis une meilleure intégration et une plus grande reconnaissance de mon travail. La relation entre utilisateurs, développeurs et supérieurs hiérarchiques a été très enrichissante et conviviale. En effet, d'un point de vue développeur, le contact avec la communauté du logiciel libre permet d'enrichir sa liste de contacts techniques et le climat de confiance avec les supérieurs a été favorable aux initiatives et donc à une meilleure productivité.

Également d'un point de vue personnel, j'ai appris à gérer un projet seul, de façon organisée et autodidacte. Lors de travaux en équipe ou même simplement pour obtenir une aide, il m'a fallu énoncer clairement un problème et les questions susceptibles de m'aider à trouver la solution (selon un principe qui pourrait se résumer à "Pose mieux la question et tu connaîtras la réponse."). Ces projets m'ont permis de mettre en évidence l'importance de l'analyse d'un problème afin de le résoudre.

Enfin, malgré les diverses sources d'enrichissement précédemment évoquées et les responsabilités que l'on m'a confiées aux cours de projets divers et variés il est à noter quelques points certes plus négatifs mais certainement plus constructifs. En effet, on peut évaluer à seulement un tiers la proportion de travail effectué en équipe alors qu'au cours de ma première année du cursus Epitech, le travail se faisait quasi-exclusivement en équipe. Cela est d'autant plus regrettable que l'ambiance très décontractée aurait été propice à un travail en équipe productif et agréable.

En conclusion, cette expérience au sein de la communauté du logiciel libre et du système Linux, tous deux en pleine expansion, est un atout majeur dans beaucoup de secteurs informatiques mais au delà des compétences spécifiques acquises dans ce domaine, il est très valorisant d'avoir apporté sa contribution aux logiciels libres.

Remerciements

Je tiens à remercier : l'école pour avoir proposé cette période de stage, Georges Lando pour sa préparation au monde de l'entreprise, Nicolas Sadirac pour m'avoir permis de préparer ce cursus qui m'a permis des la première année d'exécuter un stage déjà très intéressant d'un point de vue technique, toute l'équipe éducative et administrative, les assistants. Samuel Legros pour le contact à MandrakeSoft, Warly pour m'avoir confié des projets cohérents et qui m'ont permis d'apprendre beaucoup lors de ce stage et qui a été très à l'écoute, Frédéric Bastok et Frédéric Lepied pour leur confiance, Titi pour mon initiation à MandrakeSoft, Frédéric Crozat, Laurent Montel, Dadou et Gwenole pour leur accueil, Pixel et Greg pour leurs précieux conseils et remarques, Philippe Libat et Philippe Hetroy pour leur collaboration. Je remercie bien sur tous ceux qui ont fait que Mandrakesoft restera une expérience inoubliable.



Chapitre 5

Bibliographie

- Mandrake RPM HOWTO
<http://www.mandrakelinux.com/howtos/mdk-rpm/>
- Open Source development with CVS
<http://cvsbook.red-bean.com/cvsbook.html>
- User Contributed Perl Documentation
man perl
- Bash (GNU Bourne-Again Shell)
man bash
- GPL Licence Publique Générale GNU Version 2, Juin 1991
<http://www.linux-france.org/article/these/gpl.html> (cf annexe)

Chapitre 6

Glossaire

ADSL	Asymmetrical Digital Subscriber Line. Dans cette variante, on reçoit beaucoup plus d'infos qu'on ne peut en envoyer, idéal pour consommer du Net à la maison. Terme francophone officiel : RNA (jamais utilisé).
Cooker	Version de développement de la distribution Mandrake Linux.
Distribution	Ensemble des supports de données formant tout ce dont on a besoin pour utiliser un logiciel, une application.
DMZ	DeMilitarized Zone. Région d'un intranet doté d'une sécurisation intermédiaire entre l'extérieur (dangereux, anarchique, sauvage, le Grand Ouest) et l'intérieur (paisible et civilisé, home sweet home). (D'après Sylvain Papet).
DHCP	Dynamic Host Configuration Protocol. Protocole d'attribution dynamique des adresses sur un réseau IP.
Firewall	Barrière permettant d'isoler un ordinateur d'un réseau (tout en ne le débranchant pas complètement). Pour éviter tout piratage. En général, c'est une petite machine sous Unix qui ne contient pas de données sensibles, et étant très surveillée, les machines par derrière ne l'étant pas vraiment.
FTP	File Transfert Protocol
GNOME	GNU Network Object Model Environment. C'est à la fois une environnement utilisateur une suite bureautique et une plateforme de développement.
GPL	General Public Licence. Le statut juridique des logiciels distribués "librement" à l'origine utilisé pour le projet GNU de la FSF.
HTTP	Hyper Text Transfert Protocol
KDE	K Desktop Environment. C'est à la fois une environnement utilisateur une suite bureautique et une plateforme de développement.
LAN	Local Area Network. réseau local, dont les câbles ne font pas plus de quelques centaines de mètres de long, rencontré par exemple dans les entreprises.
MCC	Mandrake Control Center.
MNF	Multi Network Firewall
Naat	Network Appliance Administration Tool
Samba	Portage Unix du partage de fichier et imprimante Windows
SNF	Single Network Firewall
Parser	Programme réalisant une analyse syntaxique d'un code source.

Package Archive regroupant un ensemble de programmes, de documentations et de fichiers de configuration. Les principaux formats de packages (convertibles via le programme *alien*) sont : “Rpm” : introduit par Redhat et utilisé en particulier par Suse et Mandrake et “Deb” : utilisée par Debian.

Système d'exploitation Ensemble des fonctions de base (mais parfois pouvant être avancées), permettant l'usage d'un ordinateur, et sans lequel rien n'est possible. C'est l'interface entre le matériel et le logiciel.