

Rapport d'alternance

Rapport d'activité du 29 août 2011 au 20 janvier 2012

Destinataires :

Alain Isnard
Françoise Adreit
Brahim Hamid

Auteur :

Guillaume Noisette

Remerciements

Je tiens à remercier dans un premier temps, cette formation pour m'avoir permis de réaliser ce stage en alternance, car j'ai pu progresser grâce à celui-ci.

Je remercie également Françoise Adreit, ma responsable de stage, pour son suivi et son aide apportée durant ce stage.

Je tiens tout particulièrement à remercier les personnes suivantes pour l'expérience enrichissante et pleine d'intérêt qu'elles m'ont fait vivre pendant cette année au sein de Cogeser :

Frédéric Berger, président directeur général, pour m'avoir accueilli au sein de sa société.

Alain Isnard, mon tuteur et directeur général de Cogeser, pour son accueil et sa compréhension qu'il m'a accordé dès mon arrivée dans l'entreprise.

Monique Touzel, Magali Meton, Wilfrid Furbeyre, Gilles Souchois, Nathalie Rolland, l'équipe de développement, pour m'avoir intégré rapidement au sein de l'entreprise, pour leur aide, leur confiance et tout le temps qu'ils m'ont consacré tout au long de ce stage.

Sandrine Gentillet, Marie-Christine Bouloc, Marie-Claire Austruy, Jean-Paul Deoux, ainsi que l'ensemble du personnel de Cogeser pour leur accueil sympathique durant cette année.

Je tiens à exprimer ma reconnaissance à ma famille et mes amis pour leur soutien et leur patience qu'ils m'ont témoigné.

SOMMAIRE

<u>INTRODUCTION.....</u>	8
<u>SITUATION</u>	9
PRESENTATION DE L'ENTREPRISE.....	9
HISTORIQUE.....	9
SITUATION ACTUELLE, SUJETS DE PREOCCUPATION, PROJETS.....	10
ACTIVITES PRINCIPALES.....	10
ORGANISATION GENERALE.....	11
ORGANIGRAMME.....	11
ANALYSE DETAILLEE DU SERVICE ET PLACE PERSONNELLE DANS CE SERVICE.....	11
<u>PRESENTATION DU TRAVAIL, DES PROBLEMES ET DE LEUR RESOLUTION</u>	12
INTRODUCTION	12
CONCEPTS A CONNAITRE : QU'EST-CE QUE LA DUCS ?.....	12
OUTILS UTILISES	13
PROGRESS SOFTWARE	13
PRESENTATION DU TRAVAIL	13
DUCS-EDI	13
PASSAGE AUX NORMES W3C.....	15
MISE EN PLACE DE REDMINE	15
OUTIL DE NOTES DE FRAIS : WEBN2F	17
CONCLUSION SUR LE TRAVAIL REALISE	18
<u>MEMOIRE.....</u>	19
COMPILATEUR PROPRIETAIRE VERSUS COMPILATEUR LIBRE : QUE CHOISIR ?	19
INTRODUCTION.....	19
ETAT DE L'ART.....	19
LOGICIEL LIBRE.....	19
LOGICIEL PROPRIETAIRE	21
LES LICENCES.....	23
LICENCE PROPRIETAIRE	23
LICENCES LIBRES.....	24
LES COMPILATEURS	30
LES LANGAGES DE PROGRAMMATION.....	30
LES COMPILATEURS.....	33
CONCLUSION	35

<u>CONCLUSION</u>	<u>36</u>
<u>NOTES</u>	<u>37</u>
<u>TABLES DES ILLUSTRATIONS</u>	<u>38</u>
<u>SOURCES DOCUMENTAIRES</u>	<u>39</u>

INTRODUCTION

Le stage que j'effectue se déroule dans le cadre d'une formation en alternance, à savoir celle de mon Master ICE (Informatique Collaborative en Entreprise). Durant ces deux années d'étude, je vais travailler au sein de la société Cogeser.

Cogeser utilise un produit de la société Progress Software avec lequel elle réalise différents logiciels, et plus particulièrement un logiciel de Paye et de Comptabilité ainsi que des portails web (WebElite et WebN2F). Or la dernière version du produit utilisé n'inclus plus le logiciel d'édition. Cogeser était alors à la recherche d'un étudiant pour un contrat d'alternance, afin de redévelopper les anciennes éditions à l'aide du logiciel Crystal Report, j'ai donc répondu à cette annonce car elle me permettait de travailler avec une équipe de développement mais aussi de découvrir de nouvelles technologies.

Dans un premier temps, je vais vous présenter l'entreprise dans laquelle se déroule mon alternance ainsi que ma place au sein de cette société. Dans un second temps, je vous expliquerai en quoi consistent les projets auxquels j'ai participé cette année. Et dans un troisième temps, je développerai mon mémoire.

SITUATION

PRESENTATION DE L'ENTREPRISE

HISTORIQUE

La société Cogeser est une société de services informatiques qui est implantée sur Toulouse depuis 1971. C'est une entreprise spécialisée dans l'informatique de gestion au service des PME (Petites et Moyennes Entreprises) et des PMI (Petites et Moyennes Industries).

La société Cogeser a acquis en 2003 la société Agrisis, spécialisé en logiciel pour les Coopératives et les Négociant Agricoles. Puis en 2006, elle rachète la société Erel (concepteur de progiciels nomade pour gérer l'ensemble des relations commerciales de l'entreprise). En 2007, l'entreprise C2J Info (Spécialiste de la conception et du développement de solutions informatiques destinées aux Organismes de Gestion Agréés) rejoint le groupe.

En septembre 2007, la société Cogeser a été rachetée. Depuis, plusieurs sociétés sont venues renforcer le Groupe, à savoir :

- Gesti Informatique (Expert en informatique de gestion, au travers de ses activités d'intégrateur DIVALTO) en 2008.
- Infonord (Filiale de Gesti Informatique, société de services et d'engineering en informatique) en 2008.
- DDO Organisation (Spécialiste de l'hébergement en mode Saas) en 2008.
- Cogeser Centre Ouest (Expert en informatique de gestion au travers de ses activités d'intégrateur Divalto) en 2008.
- Actimédia (Spécialiste en service de traitement de courrier) en 2009.
- Actigram (Expert en logiciels de gestion pour petites entreprises, agriculteurs, viticulteurs, artisans, commerçant) en 2009.
- Ida Informatique (Expert en solutions informatiques destinées aux coopératives agricoles et aux négoce agricoles ainsi qu'aux industries agroalimentaires) en 2010.
- ProGsoft (Intégrateur majeur de Divalto dans la région de Rhône-Alpes) en 2011.
- Création de Cogeser Océan Idien, pour une activité à l'île de la Réunion en 2011.
- Datalis spécialisé dans la distribution multicanal du document d'entreprise en 2012.

SITUATION ACTUELLE, SUJETS DE PREOCCUPATION, PROJETS

Avec un chiffre d'affaires consolidé en 2011/2012 de 17 millions d'euros, le Groupe compte aujourd'hui plus de 160 collaborateurs. Il a vu aussi son activité s'élargir au fil des années avec l'intégration de nouvelles sociétés qui représentent une valeur ajoutée et qui permettent de proposer de nouveaux services. La société comporte maintenant plus de mille clients et a installé ces logiciels sur plus de cinq mille sites. De plus, le groupe est maintenant présent sur 10 sites en France :

- Toulouse (Siège)
- Angoulême
- Avignon
- Versailles
- La Rochelle
- Nantes
- Grenoble
- Arques
- Marcq-en-Barœul
- Saint-Denis (Ile de la Réunion)

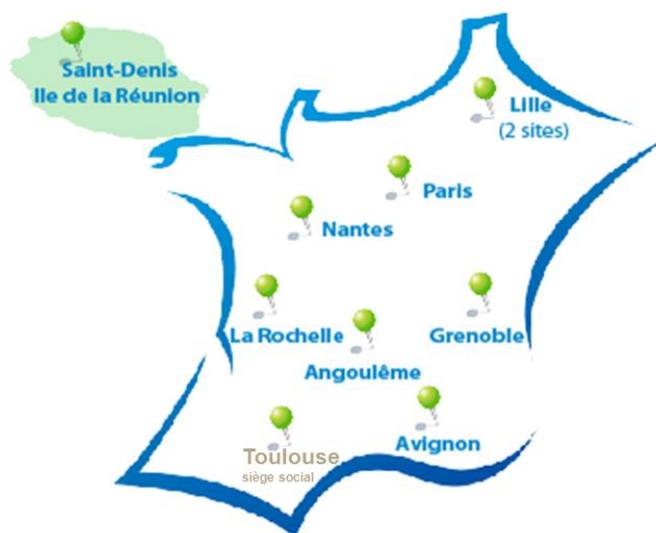


Figure 1 : Répartition des sociétés de Cogeser

ACTIVITES PRINCIPALES

Voici les principales activités dont s'occupe le Groupe Cogeser aux travers des différentes sociétés :

- L'édition de logiciels de gestion : les logiciels s'adressent aux TPE/PME/PMI, aux associations, centres de gestion, coopératives agricoles, jardineries, LISA, agriculteurs, viticulteurs, artisans, etc.
- Intégrateur majeur dans la mise en place de l'ERP Divalto, 1^{er} ERP français pour PME/PMI, du négoce, de l'industrie et des services.
- Le déploiement de matériels informatiques.
- Le déploiement de matériels de communication (système réseaux, Télécom et Téléphonie).
- La création de sites Internet et Intranet.
- Le conseil (accompagnement au changement) et la formation.
- L'hébergement en mode Saas (Software as a Service) des applications du Groupe (Divalto, Cohérence, NAIA, Erel, etc.).
- Service de traitement de courrier (mailing marketing, courrier de gestion).

ORGANIGRAMME

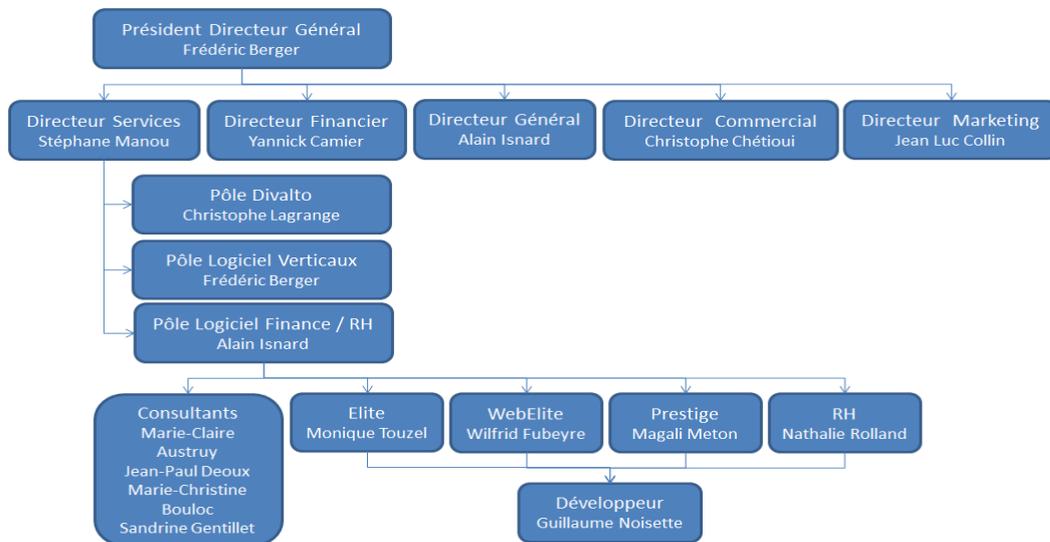


Figure 2 : Organigramme de la société

L'organisation de la société a été revue cette année, cet organigramme n'est pas complet car je n'ai pas intégré les parties dans lesquelles je n'intervenais pas.

Nous avons un Président Directeur Général en haut de l'échelle qui dirige un Directeur Général et d'autres Directeurs. Ensuite, nous avons un Directeur des Services qui gère les différents Pôles du groupe. Dans la partie service, le Pôle Divalto s'occupe d'intégrer l'ERP Divalto dans différents logiciels, le Pôle Logiciel Verticaux s'occupe de développer des logiciels pour les coopératives Agricoles, les Organismes de Gestion Agréés (OGA) et les Centres de Gestion Agréés (CGA), mais aussi des sites de e-commerces avec ProGweb. Et il y a le Pôle Logiciel Financé / Ressource Humaines qui gère notamment les logiciels Elite et Prestige, ainsi que les portails web WebElite et WebN2F. J'interviens au sein de ces quatre derniers projets, afin de maintenir ou développer différents modules.

ANALYSE DETAILLEE DU SERVICE ET PLACE PERSONNELLE DANS CE SERVICE

Le pôle Logiciel Finance/RH dans lequel je suis affecté est constitué de quatre personnes, elles gèrent le développement des progiciels de Paye (Elite, Web Elite et WEBN2F) et de Comptabilité (Prestige). Quatre personnes s'occupent de l'installation, de la formation et de l'assistance sur ces progiciels. Les quatre personnes gérant le développement se répartissent le travail en fonction de leurs connaissances. Une personne s'occupe des problèmes techniques, de l'interface d'Elite et du portail web (WebElite), une autre personne s'occupe de tout ce qui concerne la gestion des temps et les ressources humaines d'Elite ainsi que de WEBN2F, une autre de tout ce qui concerne la paye du logiciel Elite, une personne s'occupe de toute la partie comptabilité du progiciel Prestige. Pour ma part, j'apporte une aide au développement et à l'assistance surtout sur la partie Elite, Web Elite et WEBN2F. Cette équipe est dirigée par Alain Isnard qui est mon responsable de stage.

PRESENTATION DU TRAVAIL, DES PROBLEMES ET DE LEUR RESOLUTION

INTRODUCTION

Durant ma première année d'alternance, j'ai fini de reprendre toutes les éditions du logiciel Elite, cette année je réaliserai celles du logiciel Prestige. Pour attendant ce moment-là, j'ai travaillé sur plusieurs autres projets : la DUCS, le passage de Web Elite et WEBN2F aux normes W3C ainsi que leur améliorations et aussi la mise en place de l'outil Redmine.

CONCEPTS A CONNAITRE : QU'EST-CE QUE LA DUCS ?

DUCS signifie Déclaration Unifiée de Cotisations Sociales. Elle permet de déclarer, sous une forme unifiée, les cotisations sociales Urssaf, Assédic/Garp, retraite complémentaire, prévoyance, congés intempéries BTP, centre de recouvrement cinéma spectacle.

Cette déclaration concerne tous les employeurs, les dates d'échéances varient cependant en fonction de la taille de l'entreprise: pour les entreprises de moins de 10 salariés, elles doivent l'effectuer tous les trimestres, alors que celles qui comptent 10 salariés ou plus doivent l'effectuer tous les mois. Cette déclaration peut être faite directement par l'entreprise ou par l'intermédiaire d'un tiers déclarant agissant pour le compte de l'entreprise : experts comptables, associations et centres de gestion agréés...

Pour effectuer la DUCS, il faut utiliser le portail: net-entreprises.fr, c'est un portail d'administration dédié aux déclarations sociales, il permet de déclarer la DUCS en y déposant un fichier contenant toutes les informations de la DUCS ou via un formulaire électronique en ligne de manière simple, gratuite et efficace.

OUTILS UTILISES

PROGRESS SOFTWARE

Le produit Progress Software est un outil qui permet de développer des applications, il ressemble au logiciel Visual Basic, et il possède lui aussi son propre langage de programmation (OpenEdge ABL) et son type de base de données, il permet aussi de se connecter à une base de données de type SQL. Pour la version 9.1^E de Progress, il est décomposé en plusieurs parties, nous avons :

- l'éditeur qui permet d'écrire des requêtes et de les exécuter sur la base de données
- l'AppBuilder qui permet de faire l'interface d'une fenêtre et d'écrire les programmes
- le Data Dictionary où nous pouvons voir toutes les tables et les champs que contient la base de données
- le Data Administration qui permet de créer la base de données et de charger des données
- le Report Builder qui est l'outil d'édition
- Progress Explorer Tool est un outil d'administration qui permet connecter la base de données, de faire le lien entre cette dernière et IIS, et bien d'autres choses que je n'ai pas encore exploitées jusque là
- et d'autres outils que je n'ai pas encore eu l'occasion d'utiliser

Dans la nouvelle version 10.2B, nous avons encore tous les outils sauf Report Builder, cela nous a donc obligé à trouver un autre logiciel pour réaliser les éditions et par conséquent nous devons reprendre toutes les éditions. Pour cette version, Progress nous propose plusieurs produits, j'ai choisi d'utiliser : OpenEdge Architect car il s'appuie sur l'environnement de développement Eclipse que j'ai pu utiliser à maintes reprises durant ma formation, et donc que je connais bien. De plus, l'environnement Eclipse est très robuste, et grâce à sa forte utilisation beaucoup de plugins ont été développés.

PRESENTATION DU TRAVAIL

DUCS-EDI

PRESENTATION

Cogeser souhaite mettre en place la DUCS dite EDI (Echange de Données Informatisées), afin que les clients puissent réaliser plus facilement la DUCS sans avoir à tout saisir manuellement sur le portail net-entreprise. En effet, la DUCS EDI consiste générer un fichier que l'on appelle EDI contenant toutes les informations sur la déclaration, et de le déposer ensuite sur le portail net-entreprise.

REALISATION

Nous utilisons la version 4.2 de la norme DUCS EDI au format COPAYM pour la génération du fichier EDI (il existe 2 formats : COPAYM et COPLAT). Les détails de la norme sont fournis par le portail net-entreprise.net.

Grâce au cahier des charges et aux exemples de fichiers EDI, j'ai pu réaliser une procédure qui génère directement ce fichier. La partie interface des programmes et le calcul des montants pour la DUCS avaient déjà été réalisés, j'ai donc procédé à la mise en forme des données afin de les faire correspondre avec la norme.

J'ai dans un premier temps réalisé la partie DUCS EDI pour les Assedic. Afin de vérifier que le fichier correspond bien à la norme, j'ai trouvé et utilisé un site Internet qui permet de vérifier la conformité du fichier à savoir fichierdirect.declaration.urssaf.fr/seven/. A l'aide de ce site, j'ai corrigé quelques erreurs qui restaient sur le fichier EDI due aux mauvaises compréhensions du cahier des charges. Cependant, à la fin des tests certaines erreurs persistaient, je suis toujours en attente des réponses du support du site afin de les comprendre et les résoudre.

Une fois arrivée à cette étape, j'ai réalisé la partie DUCS EDI pour l'Agirc / Arcco (les caisses de retraites cadre et non cadre). J'ai réussi à trouver toute une procédure de test sur leur site qui permet dans un premier temps de vérifier les fichiers EDI générés à partir des scénarios de test qu'ils nous fournissent, en envoyant les fichiers EDI à un serveur de test. La deuxième partie consiste à faire des tests en réel avec un client qui utilisera la DUCS EDI pour faire ses déclarations. A l'heure actuelle, je suis à la fin de la première partie. Grâce à leur procédure de test qui est complète cela permet d'ajouter de la rigueur dans la génération des fichiers EDI qui n'était pas tout à fait présente dans la procédure fournie par les URSSAF.

Pour ce qui est de la partie DUCS EDI des CCPBTP, on m'a demandé de ne pas la réaliser.

```
UNA:.. ? '
UNB+UNOC:2+999888777XXXX1:5+78025472800022:5+100415:0922+00000000000188+++++
UNH+000000000000263+COPAYM:0:4.2:RT'
BGM+901:ZZZ+XXXXXXXXXXXXXXXXXXXXXXXXXXXX++
GIS+912:ZZZ
NAD+DT:999888777XXXX1:100:107+XXXXXXXXXXXXXXXXXXXX+XXXXX XXXXXXXXXXXX+XXXXXXXXXX-XXXXXXXX++08000'
CTA++:F.XXXXX'
COM+XX XX XX XX XX:TE'
NAD+HR:78025472800022:100:107+URSSAF XXX XXXXXXXX+XX XXX XXX XXXXXXXX+XXXXXXXXXXXX XXXXXXXX++08000'
NAD+FR:999888777XXXX1:100:107+XXXXXXXXXX XXXXXXXXXXXXX+XXXXXXXXXXXX XXXXXXXX+XXXXX++51100'
CTA++:XXXXX'
COM+XXXXXXXXXX:TE'
COM+XXXXXXXXXX:FX'
COM+X.XXXXX@XXXXXXXXXX-XXXXXXXXXXXX.XX:EM'
NAD+AE:999888777XXXX1:100:107+XXXXXXXXXX XXXXXXXXXXXXX+XXX XXXXXXX XXXXXXXX++51100'
CTA++:XXXXX'
COM+XXXXXXXXXX:TE'
COM+XXXXXXXXXX:FX'
COM+X.XXXXX@XXXXXXXXXX-XXXXXXXXXXXX.XX:EM'
DTM+273:1010:900'
RFF+AUM:XXXXX'
RFF+AUN:XXXXX XXXX:8.30'
NAD+WP:999888777XXXX1:100:107+XX XXXXXX XXXXXXXX+XX XXX XXXX XXXXXXXX+XXXXXXXXXXXX-XXXXXXXX++08000'
DTM+900:20100228:102'
MOA+134:393:EUR'
QTY+911:1'
QTY+912:1'
RFF+AFN:999AAA77766655441'
COT+905+100:ZZZ'
PCD+905:00000000'
MOA+920:2502:EUR'
MOA+921:2502:EUR'
QTY+912:1'
COT+905+260:ZZZ'
MOA+920:2436:EUR'
MOA+921:0:EUR'
QTY+912:1'
COT+905+671:ZZZ'
MOA+920:0:EUR'
MOA+921:703:EUR'
QTY+912:1'
UN5+S'
MOA+277:393:EUR'
PAI+:Z10'
FII+OR+XXXXXXXXXXXXXXX:XXXXXXXX XXXXXXX+XXXXX:25:108:XXXXX:25:108'
MOA+11:393:EUR'
RFF+CR:URSSAF 1010'
DTM+138:20100415:102'
UNT+47+00000000000263'
UNZ+1+00000000000188'
```

Figure 3 : Exemple de fichier EDI

PASSAGE AUX NORMES W3C

PRESENTATION

Nous souhaitons rendre les portails web, Web Elite et WebN2F, compatibles multi-navigateurs car nous ne pouvions l'utiliser que sur Internet Explorer. Pour réaliser cela, j'ai proposé de les rendre directement valide aux normes du W3C qui sont des références en matière de conformités de pages web. Cependant, suivant certains navigateurs, les ajustements ne sont pas les mêmes, en effet, tous les navigateurs n'interprètent pas de la même façon les styles appliqués aux balises. J'ai réalisé les tests d'implémentation avec les navigateurs suivants : Mozilla Firefox, Google Chrome, Safari, Opera et toujours Internet Explorer pour vérifier la compatibilité. La plus grosse partie a été d'enlever les frames car ce n'est pas valide W3C et car c'était une ancienne façon de développer.

REALISATION

Pour la réalisation de ce projet, j'ai surtout utilisé Mozilla Firefox avec les plugins Firebug-FireCookie et Html Validator pour la correction de page web car c'est un des navigateurs qui respecte le plus les règles du W3C. Je suis donc passer dans toutes les pages HTML du projet Web Elite et WebN2F afin de corriger les différents bugs que ce soit au niveau de balises mal fermés ou du style des pages qui n'étaient pas interprétés de la même façon. J'ai apporté plus de rigueur au niveau des champs affichés pour qu'ils puissent avoir le même style d'une page à une autre, mais j'ai aussi amélioré et allégé le code JavaScript qui était utilisé dans les pages. Pour porter à bien ce projet, cela m'a pris environ 2 mois de travail, ce qui a par la même occasion donner lieu à une nouvelle version de deux portails.

Maintenant une grande partie des portails Web Elite et WebN2F sont valides W3C et sont compatibles tous navigateurs, mais il reste encore 6 pages web à refaire car le code qui est présent dans ces pages n'est pas très lisible et utilise une méthode qui n'est pas très appropriée.

MISE EN PLACE DE REDMINE

PRESENTATION

Afin de mieux gérer les demandes de développement, les corrections de bugs ou les demandes d'assistance, j'ai proposé la mise en place de l'outil Redmine, que j'ai pu utiliser lors du développement de plusieurs projets que j'ai trouvé simple et complet. Redmine est un outil de gestion de projet, il permet la création de demandes comme dis plus haut, mais aussi de suivre l'état du projet grâce un diagramme de Gant ou un calendrier. On peut trouver dans cet outil, un Wiki pour y placer différentes informations et une option pour suivre l'évolution d'un repository. C'est un outil qui peut être encore amélioré en y intégrant des plugins développés par la communauté.

REALISATION

Dans un premier temps, je voulais mettre en place Redmine directement sur un serveur Microsoft 2003. Cependant, il y avait des concurrences sur certains ports, j'ai donc changé d'avis et j'ai décidé de le mettre en place sur une Virtual-box en Ubuntu Server 11.10. Cette méthode permet de déplacer plus facilement le serveur, ce qui peut être notre cas car pour l'instant Redmine a été placé sur un serveur de test.

Après avoir tout paramétré pour que Redmine fonctionne correctement, nous pouvons maintenant y accéder mais pour le moment seulement en local. A parti de là, j'ai pu créer deux projets Elite et Web Elite – WebN2F et les rattacher à leur repository, ainsi que créer les rôles dont nous aurons besoin comme le Client ou l'Administrateur ainsi que quelques comptes utilisateurs.

Redmine a été mis en place afin de voir si nous pouvions changer le site d'assistance qui a été développé il y a quelques années, mais pas seulement, en effet cet outil permet aussi de mieux gérer les projets et savoir ainsi où nous en sommes au niveau des développements. Nous pouvons aussi retrouver les anciennes demandes et ainsi résoudre des problèmes déjà rencontrés plus rapidement.

Pour le moment, Redmine est opérationnel mais il n'est pas encore utilisé.

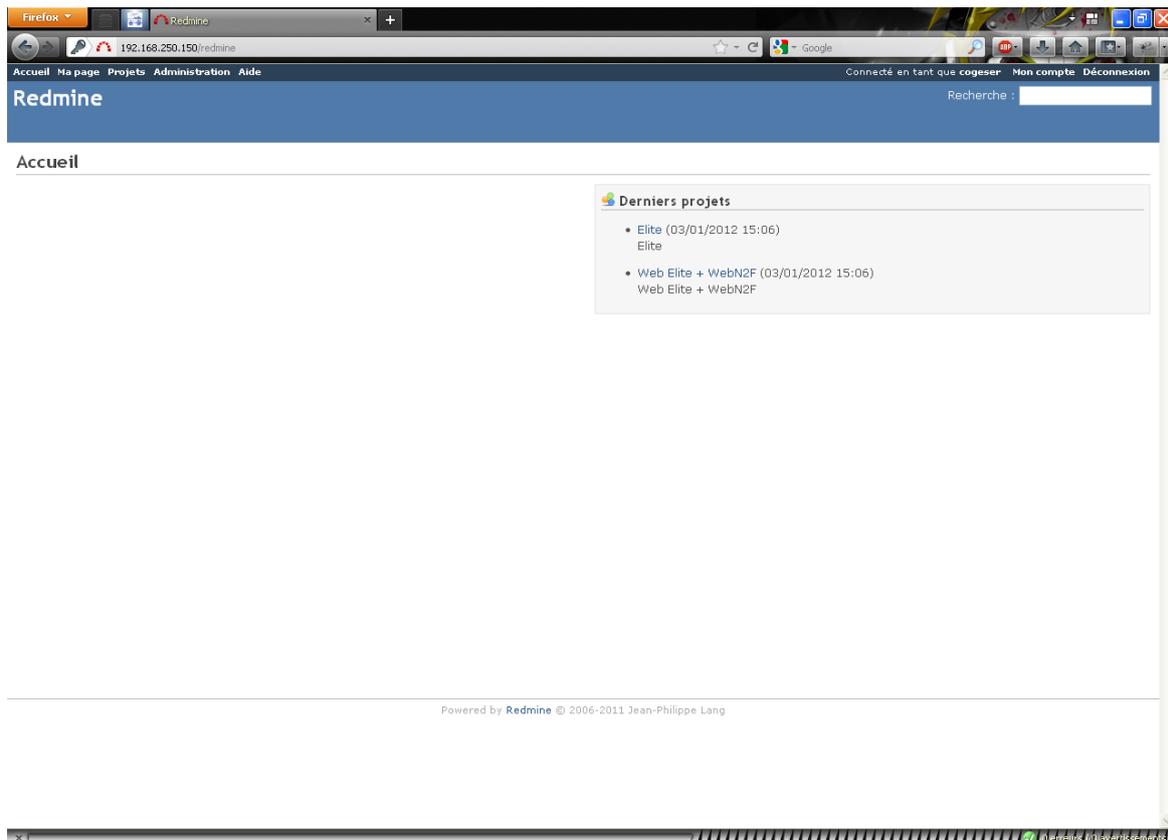


Figure 4 : Redmine

PRESENTATION

Cogeser souhaitait développer un nouvel outil de saisie et contrôle des notes de frais. Le travail a commencé à être développé par une personne qui a quitté la société. J'ai donc repris son travail afin de le finaliser en le corrigeant et en apportant de nouvelles fonctionnalités afin de le rendre commercialisable.

Le produit était divisé en cinq grandes parties :

- Les paramètres généraux où le client définissait la nature de frais (par exemple : restauration midi de type réel plafonné à 10€, frais kilométrique, ...), les barèmes kilométriques et bien d'autres choses.

- La partie saisie des frais qui se divisait en deux étapes, la définition générale de la note de frais (saisie d'une description, saisie des dates car cela peut se faire toutes les quinze jours, mensuellement, ...) et ensuite, vient la saisie des frais (par exemple : le repas du midi est de type restauration midi, il a été pris le 01/06/12 et son prix s'élève à 15€) et automatique on calcule le montant à rembourser en fonction de ce que l'on a noté dans la nature du frais qui est rattaché (ici ça serait 10€).

- Le contrôle de saisie des frais s'effectue si dans les paramètres généraux nous avons défini un nombre de niveaux de contrôle de 2. Une (ou plusieurs) personne(s) s'occupe de contrôler la note de frais et de dire si elle est valide ou alors la personne dit que tel ou tel frais n'est pas valide.

- La partie validation est active si le nombre de niveaux de contrôle est d'au moins 1. Cette section permet d'envoyer la note de frais ou non à la partie règlement qui peut être soit par virement soit par export vers notre logiciel de paye.

- Dans la partie virement, on définit une date de traitement de la note de frais, un libellé et la banque et cela nous donne un fichier à envoyé à la banque sélectionnée pour qu'elle traite les virements. Dans l'export vers la paye, on définit seulement un mois et une année de traitement car le remboursement se déroulera lors de la paye du salarié et apparaîtra sur son bulletin de salaire.

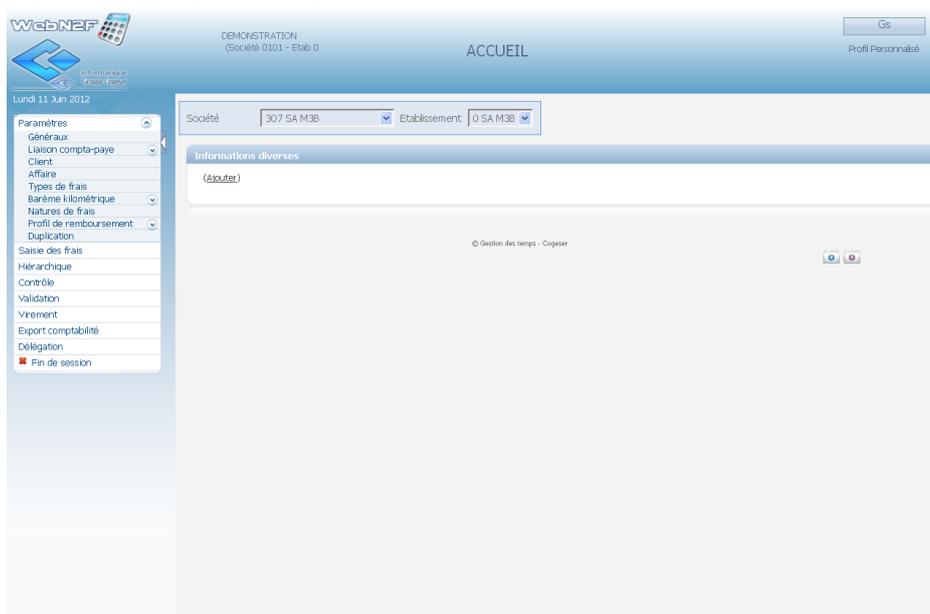


Figure 5 : WEBN2F

REALISATION

J'ai repris cet outil afin de le compléter en corrigeant les différents bugs apparents et en apportant des fonctionnalités que nos futurs clients attendaient comme lors de la saisie d'un frais, nous puissions ajouter un justificatif, des invités internes ou externes de la société afin d'adapter le remboursement. De plus, j'ai ajouté l'export vers la comptabilité qui était une fonction demandée mais qui n'était pas encore implémentée avec la prise en compte de la partie analytique pour chaque salarié.

Après avoir réalisé ces améliorations et bien d'autres du produit, un formateur et moi-même sommes allés voir un client qui était très intéressé par ce produit afin de voir si cela correspondait à ses attentes. Ce produit correspondait, en effet, à ses attentes mais il manquait encore des choses telles que la modification du montant remboursé par les différents niveaux de sécurité, un nouveau niveau de validation pour un supérieur hiérarchique, et aussi, un système permettant de définir une personne de confiance pour effectuer nos tâches de validation ou de contrôle lorsque nous ne sommes plus présents au sein de l'entreprise.

Suite à toutes ces propositions, nous les avons acceptés et mis en place sur notre produit sans trop de difficultés. Maintenant, le produit a été acheté et mis en place chez ce client.

CONCLUSION SUR LE TRAVAIL REALISE

Au niveau de la DUCS EDI, le travail arrive à son terme, en effet en traitant la partie Agirc / Arrco cela permet d'être sûr que le fichier EDI est généré correctement grâce à leur procédure de test qui est complète. Il reste maintenant à effectuer des tests avec un client réel.

Pour le passage aux normes W3C, cela est fini, les quelques pages qui sont à reprendre le sont car des corrections ont été apportées sur l'ancienne version, et je devrais réaliser le portage prochainement.

Redmine va subir des tests au sein du service de Web Elite dans un premier temps. Pour voir s'il correspond bien aux différents besoins du service, et si c'est le cas il sera mis en place sur un serveur de production.

Le produit WEBN2F est actuellement mis en place chez un seul client, il est donc en « test réel » et risque de subir encore quelques petites modifications.

MEMOIRE

Compilateur propriétaire VERSUS compilateur libre : QUE CHOISIR ?

INTRODUCTION

J'ai choisi ce sujet car ma première mission au sein de Cogeser était de réaliser toutes les éditions qui étaient faites avec l'outil Report Builder de Progress Software avec Crystal Report puisque Progress a abandonné son outil.

Dans ce mémoire, je vais dans un premier temps vous présenter la différence entre les logiciels libres et les logiciels propriétaires, puis dans un second temps, je présenterai les quelques licences qui s'appliquent à ces logiciels. Pour terminer, je vous exposerai des compilateurs libres et propriétaires.

ETAT DE L'ART

LOGICIEL LIBRE

Au commencement de l'informatique tout était libre car elle n'était utilisée que par des professionnels et des chercheurs académiques. Ils y partageaient leurs connaissances pointues qui étaient des codes sources, dans le but de développer des logiciels plus rapidement. Puis le code source fut caché par les entreprises à leurs utilisateurs, notamment grâce aux avancées technologiques comme le Personal Computer, Unix et Windows. Dans le même temps, la séparation progressive du « langage humain » par rapport aux instructions exécutées sur la machine ont joué un rôle important dans cette réalisation.

A l'heure actuelle, la séparation entre le langage de programmation et langage machine est de plus en plus marquée grâce aux progrès scientifiques réalisés durant ces cinquante dernières années. Ce fut l'apparition des logiciels propriétaires.

D'après la Free Software Foundation (FSF¹) qui est une association fondée par Richard Stallman, un logiciel est libre s'il confère à son utilisateur quatre libertés:

0. La liberté d'exécuter le programme, pour tous les usages,
1. La liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins,
2. La liberté de redistribuer des copies du programme,
3. La liberté d'améliorer le programme et de distribuer ces améliorations au public, pour en faire profiter toute la communauté.

Les logiciels libres ne sont pas réalisés par une unique entreprise, ils sont faits par toute une communauté. Tout le monde peut apporter sa contribution par exemple en rapportant des bugs, en suggérant des idées ou même en programmant. Une règle importante dans le logiciel libre est que tout le monde est égal et possède les mêmes droits, que ce soit les entreprises, les développeurs et même les utilisateurs.

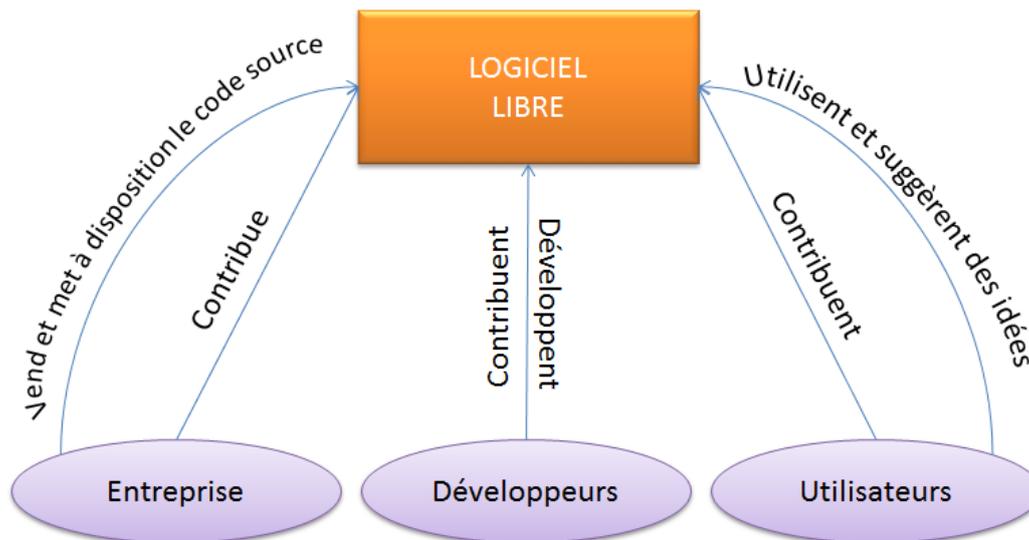


Figure 6 : Le logiciel libre

Une personne qui utilise un logiciel libre possède le droit de modification et d'utilisation à titre privé, et si elle souhaite publier des modifications qu'elle apporte, elle n'a pas l'obligation de prévenir quelqu'un en particulier ou de le faire d'une manière particulière.

De même, la liberté d'utilisation d'un logiciel se traduit par le fait que n'importe qui ou n'importe quelle entité puisse l'utiliser sur n'importe quel système et cela pour n'importe quelle tâche sans être obligé de communiquer ce qui est fait.

Si l'utilisateur souhaite utiliser le droit qui lui permet de redistribuer l'application, il devra fournir le code source de l'application et les formes binaires ou exécutables du programme, sauf si le langage utilisé ne le permet pas. Si le code source fourni est illisible, cela ne peut être considéré comme un vrai code source et dans ce cas il y a non-respect des libertés 1 et 3.

Si le programme qui vous a été livré est conçu pour exécuter les versions modifiées par un autre personne et refuse vos modifications, alors ce programme entrave la liberté 1 car les binaires ne sont pas libres, même si c'est le cas pour le code source qui a servi à les compiler. Cette pratique est appelée « tivoïsation », « verrouillage » ou encore « secure boot ».

Le terme de « logiciel libre » ne signifie pas qu'il n'est pas commercialisable, au contraire il doit permettre l'usage, le développement et la distribution commerciale. En effet, on peut très bien se procurer un logiciel libre en le payant mais aussi gratuitement. Son acquisition n'empêche pas de le copier, de le modifier et même d'en vendre des copies de ce logiciel.

Une licence d'un logiciel libre peut tout de même imposer quelques restrictions comme changer le nom de la nouvelle version, enlever un logo ou marquer les modifications que l'on a produites comme étant de notre fait. Cependant les contraintes ne doivent pas

limiter la distribution du nouveau logiciel libre modifié. Une licence peut nous imposer d'envoyer une copie au(x) développeur(s) d'origine.

Afin de déterminer si un logiciel et donc sa licence, correspondent aux critères « libres », il faut beaucoup de réflexion, de vérification et parfois même l'aide d'un juriste.

Pour déterminer si un langage est libre, il faut se poser la question suivante : le compilateur est-il libre ? Pour le savoir, il faut regarder sa licence et les différents termes présents dans le contrat du logiciel, s'ils correspondent aux quatre libertés énoncées ci-dessus, c'est un logiciel libre. Dans le cas contraire, il est propriétaire ou les termes du contrat sont mal écrits et il faut alors l'analyser avec un professionnel.

Les langages libres sont les plus présents sur le marché informatique car ils sont maintenus par les personnes qui les ont développés et par une communauté. De plus, ils sont généralement gratuits et faciles à se procurer. Les langages de programmation libres les plus courants sont le C (1972) / C++ (1983), Java (1995) et Ruby (1995).

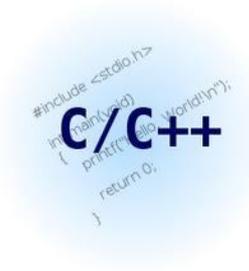


Figure 7 : Langage C/C++



Figure 8 : Langage Java

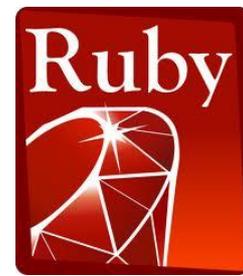


Figure 9 : Langage Ruby

LOGICIEL PROPRIETAIRE

Un logiciel est dit propriétaire ou privé s'il possède un propriétaire ou une entité qui entrave l'accès à une des lois des logiciels libres énoncées par la FSF. En général, les auteurs de ce genre de logiciel souhaitent garder le contrôle sur les droits de propriété et d'usage du logiciel. L'utilisateur final de celui-ci doit généralement accepter un Contrat de Licence Utilisateur Final (CLUF), de ne faire aucune copie et de ne pas retranscrire même partiellement le logiciel. Du fait qu'un auteur ne divulgue que très rarement le code source de son logiciel propriétaire cela permet une certaine sécurité à l'utilisateur. En effet, les failles de sécurité sont plus difficiles à trouver pour les hackers. Il peut persister des problèmes : si le logiciel plante l'utilisateur ne pourra pas analyser le code puisqu'il n'a pas accès au code source. Il devra donc attendre que l'entreprise corrige le bug.

De plus, les logiciels propriétaires sont soumis aux mises à jour des développeurs qui sont les seuls à pouvoir les améliorer. Il existe aussi des risques liés à l'intégration du fait du choix du langage utilisé par l'auteur, le langage est la plupart du temps considéré comme propriétaire. Si un utilisateur choisit de faire des copies pour d'autres usagers, alors que cela lui est interdit d'après le contrat qu'il a signé lors de l'acquisition du logiciel, alors ce dernier devient ce que l'on appelle couramment un « pirate ».

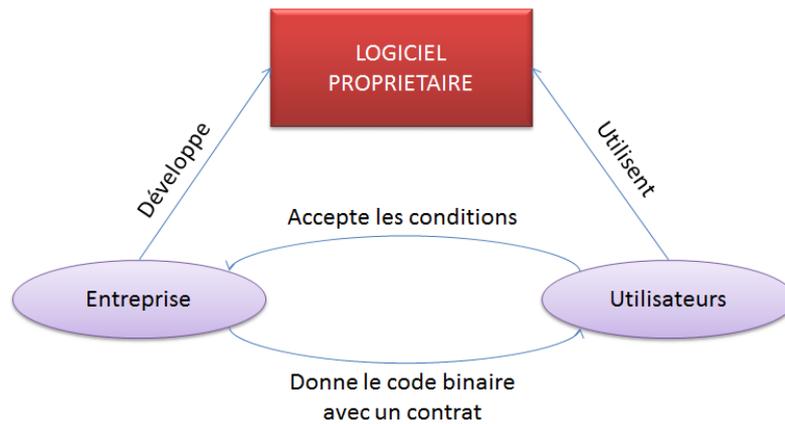


Figure 10 : Le logiciel propriétaire

Pendant en France, il y a certaines utilisations que l'auteur ne peut pas interdire :

- Les représentations privées et gratuites effectuées exclusivement dans un cercle de famille.
- Les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective.
- Les analyses et courtes citations (sous réserve que soient indiqués clairement le nom de l'auteur et sa source) justifiées par le caractère critique, polémique, pédagogique, scientifique ou d'information de l'œuvre à laquelle elles sont incorporées.
- Les copies de revues de presse.
- La diffusion des discours prononcés en public.
- Les reproductions d'œuvre d'art destinées à figurer dans un catalogue de vente aux enchères faite par un officier public ou ministériel.
- La parodie, la pastiche et la caricature.
- Les actes nécessaires à l'accès au contenu d'une base de données électronique pour les besoins et dans les limites de l'utilisation prévue au contrat.
- L'enseignement et la recherche à certaines conditions.
- La reproduction et la représentation en vue de la consultation personnelle de personnes handicapées (par exemple la transcription en braille).

Mais ces différentes exceptions ne doivent pas porter atteinte à l'exploitation normale du logiciel, ni causer un préjudice injustifié aux intérêts légitimes de l'auteur, comme le précise la loi relative au droit d'auteur et aux droits voisins dans la société de l'information (DADVSI).

Un langage est propriétaire si nous devons utiliser un logiciel particulier pour consulter, éditer, modifier ou exécuter le code source ou binaire. Cela revient à dire que lors du développement, nous devons utiliser un logiciel propriétaire ou un de ses packages pour faire avancer notre logiciel ou pour l'exécuter. Si le logiciel utilisé pour réaliser tout cela disparaît, toute l'application qui utilise ce langage disparaîtra aussi si elle est développée entièrement avec celui-ci.

Les langages propriétaires sont quasiment autant présents que les langages libres. Certains de ces langages sont sur le marché de l'informatique depuis plusieurs dizaines d'années comme Visual Basic (Microsoft en 1991), ActionScript (Adobe en 1998) ou encore Advanced Business Language (ABL - Progress Software en 1984).



Figure 11 : Visual Basic (Microsoft)



Figure 12 :
ActionScript (Adobe)

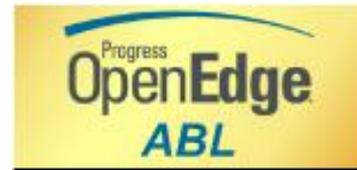


Figure 13 : Advanced Business Language
(Progress Software)

LES LICENCES

Il existe plusieurs types de licence qui permettent de déterminer si un logiciel est propriétaire ou libre. Parfois on peut faire l'amalgame sur certaines licences que l'on pense libre et qui sont en réalité propriétaire. C'est pour cela que j'ai souhaitais vous présenter plusieurs d'entre elles.

LICENCE PROPRIETAIRE

Ce type de licence est aussi appelé licence privative. Elle correspond à toute licence qui n'est pas libre, c'est-à-dire si elle entrave une des quatre libertés : utiliser, étudier, dupliquer, modifier et distribuer. Généralement, ce type de licence se fait par un contrat de licence utilisateur final où il est écrit les différentes restrictions. L'achat de ce genre de licence ne correspond pas à l'achat du logiciel, mais seulement l'acquisition du droit d'utilisation limité au cadre établi par la licence. Le transfert de propriété n'existe dans ce cas présent que s'il y a vente du droit d'auteur. L'utilisateur doit donc consentir à l'interdiction de partage ou d'utilisation définie par cette licence. De plus, l'auteur peut très bien imposer à l'utilisateur l'interdiction d'effectuer de la rétro-ingénierie². On peut souvent croiser ce genre de clause dans les licences propriétaires, mais dans certains pays ces clauses sont nulles car la loi de ces pays ne les permettent pas. Mais le renoncement à ces clauses nécessite un accord explicite, c'est pour cela que l'on emploie pas le terme de licence mais de contrat de licence.

Cependant, toute cette privation des libertés n'est pas forcément dues aux licences, parfois ce sont les lois de la propriété intellectuelle comme le droit d'auteur ou le brevet qui confèrent un monopole d'utilisation aux titulaires de ces privilèges.

Généralement, les personnes qui souhaitent utiliser ce type de licence emploient le terme de « commercial », alors qu'un logiciel libre peut aussi se vendre, tout comme un logiciel propriétaire peut être gratuit. On remarque donc elles n'ont pas étudié correctement les différents types de licence ou alors qu'elles n'adhèrent pas aux principes du libre.

LICENCES LIBRES

Il existe énormément de licences libres, donc nous allons ici en voir seulement quelques-unes, celle que je considère comme les plus connues. Mais dans un premier temps, je vais vous expliquer ce que signifie le terme copyleft qui est très employé dans le monde du libre.

COPYLEFT

C'est une méthode qui permet de rendre libre un programme et aussi toutes ses versions qui modifiées ou étendues. Le copyleft permet de garantir les libertés du logiciel libre. Un exemple de non utilisation du copyleft : si l'on met notre logiciel libre dans le domaine public et donc que l'on rend notre logiciel libre, cela ne le protège pas car une personne peut le récupérer, y effectuer des modifications et le rendre propriétaire et le distribuer à des utilisateurs qui eux n'auront plus de libertés.

Le copyleft permet d'encourager les programmeurs à augmenter le nombre de logiciel libre. Cela permet aux programmeurs, qui se trouvent dans des entreprises ou des universités qui cherchent à toujours gagner plus d'argent, de contribuer au logiciel libre par des améliorations à obtenir la permission de le faire.

Pour mettre un logiciel en copyleft, il faut dans un premier temps le mettre sous copyright, ensuite il faut ajouter des conditions de distribution, qui sont un outil juridique donnant à chacun le droit d'utiliser, de modifier et de redistribuer le code du programme, ou tous les programmes qui en sont dérivés, mais seulement si les conditions de distribution demeurent inchangées. On remarque donc que le copyleft à l'inverse du copyright, qui lui restreint les libertés des utilisateurs, permet de les garantir.

Le copyleft n'est qu'un concept général, il ne peut pas s'appliquer seul, il faut donc l'associer à une licence lorsque l'on souhaite l'utiliser et vérifier que cette licence accepte bien le copyleft.

GNU GENERAL PUBLIC LICENSE (GPL)

Il s'agit d'une licence de logiciel libre et d'un copyleft qui a été créé par la Free Software Foundation. Cette licence est utilisée dans le projet GNU et elle correspond à LA référence des licences de logiciel libre. Richard Stallman et Eben Moglen ont été les deux premiers rédacteurs de cette licence. Le but principal de ces auteurs était de créer une licence unique qui pourrait être utilisée pour chaque projet afin de partager leur code source. A l'heure actuelle, nous sommes à la troisième version de cette licence (publiée le 29 juin 2007). La GNU GPL permet à toute personne de recevoir une copie d'un travail sous cette licence. Chaque personne qui adhère aux termes et aux conditions de la GPL a la permission de modifier le travail, de l'étudier et de redistribuer le travail ou un travail dérivé. Cette personne peut également percevoir de l'argent pour ce service ou pas en sachant que la personne qui a acheté ce programme peut le distribuer gratuitement juste après. Ce dernier point distingue la GPL des autres licences de logiciels qui interdisent la redistribution

dans un but commercial. En cas de modification, le résultat doit être placé sous la même licence.

La GPL a été conçue en tant que licence non pas comme un contrat. Au niveau juridique, cela change énormément les choses, les contrats relèvent du droit des contrats alors que les licences relèvent eux des droits d'auteur. Cependant dans certains pays cette distinction n'est pas faite.

Les principes appliqués par l'auteur de la GPL sont les suivants :

- Le droit d'auteur interdit la copie, la distribution et les travaux dérivés, sans permission de l'auteur.
- Une violation d'une de ces interdictions n'est pas une violation de licence, mais une violation de la loi.
- À travers la GPL, l'auteur donne unilatéralement des permissions, soigneusement limitées, mais sans contrepartie.
- Ne pas respecter une des clauses de la GPL revient donc à commettre sans permission un acte interdit par le droit d'auteur, donc à violer la loi.
- Comme aucune contrepartie n'est demandée à l'utilisateur, les logiciels sous GPL peuvent en principe être installés sans accepter les termes de la licence. En effet, l'auteur ne demande rien à l'utilisateur, ce dernier n'a donc rien à accepter.

Bien que l'utilisateur n'a rien à accepter, il doit faire attention au fait que les permissions qui lui sont accordées à travers la GPL sont limitées.

La section 7 de la licence GNU GPL est appelé « la liberté ou la mort » car elle stipule que les obligations imposées par les décisions de justice ou les allégations de contrefaçon de brevet ne libèrent pas les utilisateurs des conditions de la licence. Et donc s'ils ne peuvent pas satisfaire à la fois les obligations imposées et celles de la GPL, ils doivent arrêter la distribution du programme. L'application réelle de la licence GPL est complexe car les intérêts de chaque partie et les moyens de rémunération le sont.

Pour en revenir au texte de la licence GNU GPL, il est sous copyright de FSF, il n'est pas libre puisque on a le droit de le copier et le distribuer mais tout changement est interdit.

Cette licence est utilisée pour le compilateur C/C++ : GNU Compiler Collection (GCC), mais aussi pour le langage Ruby et son interpréteur.



Figure 14 : GNU General Public License

GNU LESSER GENERAL PUBLIC LICENSE (LGPL)

Cette licence est une licence de logiciel libre rédigé par la Free Software Foundation, mais qui n'a pas un copyleft fort, car elle permet de lier des modules libres à des non libres. Elle est compatible avec la GPLv3. C'est une licence qui s'applique souvent aux bibliothèques. La seule contrainte qu'émet cette licence est que toute modification apportée à la bibliothèque doit être sous licence LGPL. A part ces deux points cette licence est équivalente à la licence GPL.



Figure 15 : GNU Lesser General Public License

Il existe aussi une autre version plus complète de GNU GPL qui comprend les logiciels utilisés sur le réseau qui est GNU Affero General Public License (AGPL).



Figure 16 : GNU Affero General Public License

GNU ALL-PERMISSIVE LICENSE

C'est une licence simple et permissive, qui peut être utilisée avec la GNU GPL. Elle est généralement utilisée pour les fichiers README, INSTALL ou des petits fichiers d'aide.

Copying and distribution of this file, with or without modification, are permitted in any medium without royalty provided the copyright notice and this notice are preserved. This file is offered as-is, without any warranty.

Figure 17 : GNU All-Permissive License

LICENCE CC0

CC0 est une licence Creative Commons dédiée au domaine public. Un logiciel qui utilise cette licence est mis dans le domaine public dans les limites les plus larges permises par la loi. Si cela n'est pas possible, CC0 fournit aussi une licence simple et permissive comme substitut. Cette licence est compatible avec la GNU GPL. Elle est par exemple utilisée pour le site <http://www.gnu.org>.



Figure 18 : Licence CC0

LICENCE CECILL

C'est un acronyme pour Ce(A)C(NRS)I(NRIA)L(ogiciel)L(ibre), elle est compatible GNU GPL. Je parlerai ici que de la version « standard », pas des versions CeCILL-B ou CeCILL-C. L'acceptation de la licence CeCILL se fait dès le chargement ou dès le premier exercice par le licencié de l'un des droits concédés par le contrat sur le logiciel portant cette licence. Après cette acception, les utilisateurs sont tenus de préserver le statut de logiciel libre aux versions qu'ils redistribueront et d'inscrire leurs noms ainsi que les dates de création s'ils ont apporté une contribution au logiciel. De plus, lors des redistributions du logiciel le prix de celui-ci ne doit ne pas excéder le coût du transfert des données. Le logiciel est soumis au respect des conditions que l'auteur a choisi lors de sa diffusion, personne d'autre ne peut modifier ces conditions. L'utilisateur en acceptant cette licence s'engage à ne pas supprimer ou modifier les mentions de propriété intellectuelle apposées sur le logiciel et à les reproduire à l'identique sur les copies modifiées. Par ailleurs, la personne distribuant ce logiciel peut si elle le souhaite proposer des prestations d'assistance technique et de maintenance ainsi qu'une garantie mais cela n'engagera qu'elle. L'utilisateur peut, sous réserve de prouver la faute de la personne qui lui a fourni le logiciel, solliciter la réparation du préjudice qu'il subirait à cause du logiciel. Une partie de la licence CeCILL engage les développeurs de programmes à une certaine forme de coopération avec les utilisateurs. On pourrait voir ceci comme un problème pour le développeur ; cependant, si vous êtes sûr de vouloir travailler dans ce sens avec les utilisateurs, alors ce n'est pas un problème pour vous. Cette licence permet aussi d'effectuer une résiliation en cas de manquement de l'utilisateur à ces obligations mises à charge par la licence, celle-ci pourra se faire de plein droit, trente jours après une notification adressée à l'utilisateur. L'utilisateur dont le contrat est résilié ne pourra plus utiliser, modifier et distribuer le logiciel. Cette licence est régie par la loi française, et si une ou plusieurs dispositions de cette licence s'avèrent contraire à une loi ou un texte applicable, cette loi ou texte prévaudrait.



Figure 19 : Licence CeCILL

LICENCE RUBY

C'est la licence qui s'applique au langage Ruby, elle est compatible avec la GNU GPL. D'après cette licence, on peut distribuer des copies du logiciel sous forme de code source sans restriction et en mettant toutes les notices de copyright et les avertissements associées. On peut modifier le logiciel si l'on remplit plusieurs conditions tels que l'utilisation du logiciel modifié au sein de son entreprise ou organisation, renommer les exécutables non-standard de sorte que leurs noms ne sont pas incompatibles avec les exécutables standard, qui doivent également être fournis, etc... La distribution des copies modifiées est elle aussi soumise à plusieurs conditions comme donner des instructions pour obtenir le logiciel d'origine, ou encore des modalités spécifiques faite par l'auteur. Si une partie du logiciel ou des scripts soumis à cette licence sont utilisés dans un autre logiciel alors cette licence reste applicable sur ce script mais le reste du logiciel peut être soumis à une autre licence.

Ruby is copyrighted free software by Yukihiro Matsumoto <matz@netlab.co.jp>. You can redistribute it and/or modify it under either the terms of the GPL (see COPYING.txt file), or the conditions below:

1. You may make and give away verbatim copies of the source form of the software without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may modify your copy of the software in any way, provided that you do at least ONE of the following:
 - a) Place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or by allowing the author to include your modifications in the software.
 - b) Use the modified software only within your corporation or organization.
 - c) Rename any non-standard executables so the names do not conflict with standard executables, which must also be provided.
 - d) Make other distribution arrangements with the author.
3. You may distribute the software in object code or executable form, provided that you do at least ONE of the following:
 - a) Distribute the executables and library files of the software, together with instructions (in the manual page or equivalent) on where to get the original distribution.
 - b) Accompany the distribution with the machine-readable source of the software.
 - c) Give non-standard executables non-standard names, with instructions on where to get the original software distribution.
 - d) Make other distribution arrangements with the author.
4. You may modify and include the part of the software into any other software (possibly commercial). But some files in the distribution are not written by the author, so that they are not under this terms.

They are gc.c(partly), utils.c(partly), regex.[ch], st.[ch] and some files under the ./missing directory. See each file for the copying condition.
5. The scripts and library files supplied as input to or produced as output from the software do not automatically fall under the copyright of the software, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this software.
6. THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Figure 20 : Licence Ruby

LICENCE X11

C'est une licence simple et permissive, sans copyleft qui est compatible GNU GPL. Elle est parfois appelé « licence du MIT », mais c'est une erreur car le MIT publie des logiciel sous tout type de licence. Cette licence donne à toute personne recevant le logiciel le droit illimité de l'utiliser, le copier, le modifier, le fusionner, le publier, le distribuer, le vendre et de changer sa licence. La seule obligation est de mettre le nom des auteurs avec la notice de copyright. Cette licence n'est en réalité qu'un modèle, il faut donc l'adapter à notre logiciel en modifiant le Consortium, le nom du logiciel, etc...

Copyright (C) 2012 X Consortium

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

X Window System is a trademark of X Consortium, Inc.

Figure 21 : Licence X11

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL), VERSION 1.0

C'est une licence de logiciel libre créé par Sun Microsystems et elle s'applique à des outils comme l'IDE NetBeans, le serveur GlassFish. Elle n'est pas compatible avec la GNU GPL. Elle est l'une des seules à implémenter un copyleft qui ne se préoccupe pas seulement des libertés conférées par la licence dans laquelle est distribué le tout, mais de conserver toutes les licences. Si un programme est sous licence CDDL et que l'on y apporte des modifications et que ensuite l'on l'intègre à un logiciel propriétaire, le programme reste sous licence CDDL mais ne contamine pas le reste du logiciel qui peut donc rester propriétaire.

JSON LICENSE

La JSON license correspond à la licence d'origine de la mise en œuvre du format d'échange de données JSON. Cette licence se considère libre mais à cause d'une clause qui stipule que : « le logiciel sera utilisé pour le Bien, non pour le Mal », elle ne l'est pas. Cela met en place une restriction sur l'usage et donc c'est en opposition avec la liberté 0 des logiciels libres. Il est probablement impossible de forcer une application à être utilisé que pour le Bien, mais on ne peut en présumer.

Copyright (c) 2002 JSON.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Figure 22 : JSON license

LES COMPILATEURS

Il existe beaucoup de langage de programmation et donc beaucoup de compilateurs. Je ne vous en présenterai que quelques-uns, ceux que je considère comme les plus utilisés.

LES LANGAGES DE PROGRAMMATION

C

Le langage C est un langage de programmation impératif de bas niveau qui a été conçu pour la programmation système. Il a été inventé en 1970 avec la venue d'UNIX par Dennis Ritchie et Ken Thompson, puis Brian Kernighan a aidé à le populariser et a effectué quelques modifications. Ce langage est connu car il est basé sur un format libre. Beaucoup de bibliothèques logicielles et de compilateurs l'utilisent. Il met en œuvre peu de concepts ce qui permet de le maîtriser rapidement. De plus, il ne spécifie pas le comportement des fichiers exécutable afin de s'adapter aux capacités de chaque ordinateur. L'écriture de logiciels avec ce langage ne nécessite pas le besoin d'un support d'exécution comme une machine virtuelle pour Java. Le C a influencé beaucoup d'autres langages comme le C++, C#, PHP ou encore Java.

C++

Il est apparu en 1983, il a été réalisé par Bjarne Stroustrup pour améliorer le langage C. Il contient la prise en charge des classes, ainsi que de nombreuses autres fonctionnalités comme les fonctions virtuelles, la surcharge des opérateurs, l'héritage (simple ou multiple), les « templates », la gestion des exceptions, etc... Il a été appelé C++ car le ++ correspond à l'incrément de la valeur d'une variable de 1, cela signifie que C++ est un niveau au-dessus du C.

C#

Le langage de programmation C# est un langage orienté objet qui a été créé en 2001 par Microsoft et notamment par un de leur employé : Anders Hejlsberg. Il a été créé afin d'utiliser toutes les capacités qu'offre la plate-forme Microsoft .Net dont il est extrêmement dépendant. Il est très proche de Java dont il reprend la syntaxe générale et certains concepts, mais il s'inspire aussi du C++ pour la possibilité de surcharger des opérateurs. Il utilise tout comme Java une machine virtuelle : la Common Language Runtime (CLR) pour exécuter des programmes écrit en C#.

RUBY

Il est un langage de programmation libre qui est interprété, orienté objet et multi-paradigme³. Ce langage a été créé par le programmeur japonais Yukihiro Matsumoto en 1995, bien que son idée soit apparue en 1993. Il souhaitait rassembler certaines fonctionnalités de ses langages préférés de l'époque principalement Perl, Smalltalk, Eiffel, Ada et Lisp. Son nouveau langage devait astucieusement mêler la programmation impérative et fonctionnelle. Son but était « d'essayer de rendre Ruby le plus naturel possible, pas nécessairement simple ». Depuis sa première publication, Ruby intéresse de plus en plus de personne, aujourd'hui d'après l'index TIOBE il se trouve à la dixième place des langages les plus utilisés. C'est un langage complètement objet où toute donnée est un objet même les types primitifs. Il est très flexible car il permet de modifier, supprimer ou même ajouter des méthodes aux objets de base. Il possède un ramasse-miette, un système de gestion des exceptions, l'héritage simple et l'inclusion de modules dans les classes pour « l'héritage multiple ». Il peut définir des blocs qui servent d'itérateur et permettent de passer des morceaux de programme en paramètre. Ce langage possède de nombreuses autres fonctionnalités. Il existe beaucoup d'interpréteurs différents pour Ruby tels que JRuby qui est écrit en Java, IronRuby qui est un compilateur fonctionnant sur la Dynamic Language Runtime (DLR) du framework .Net, et bien d'autres interpréteurs. Il existe aussi un framework pour Ruby : Ruby on Rails qui permet de créer des applications web et qui suit le modèle MVC (Modèle-Vue-Contrôleur), il est utilisé par exemple pour l'application Redmine qui est un outil de gestion de projet.

JAVA

Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, des employés de Sun Microsystems, et avec le soutien de Bill Joy, un des cofondateurs de Sun Microsystems. Il a été présenté officiellement le 23 mai 1995. Il reprend en grande partie le langage C++ duquel certains concepts on était enlevé comme les pointeurs et l'héritage multiple, ce dernier est remplacé par les interfaces. Dans ce langage, tout est objet sauf les types primitifs tel que les nombres entiers, les nombres à virgule flottante, etc... Java permet de développer des applications dites « lourdes », qui peuvent être exécuté sur n'importe quelle machine pourvue qu'elle possède la machine virtuelle, mais aussi des applications client-serveur, côté client avec des applets et côté serveur soit des servlets ou des JavaServer Pages (JSP) en remplacement des langages comme PHP ou ASP. Ce langage a permis de développer plusieurs projets par exemple un système d'exploitation JavaOS, des environnements de développement Eclipse, des bibliothèques graphiques Swing, et beaucoup d'autres produits. Il se base sur cinq objectifs qui sont orienté objet et familier; robuste et sûr; indépendant de la machine employée pour l'exécution; très performant; interprété, multitâches et dynamique.

ACTIONSCRIPT

Il a été développé par la société Adobe Systems, il est utilisé au sein d'application tels qu'Adobe Flash, Adobe Flex, Flash media server, JRun ou encore Macromedia Generator. ActionScript est un langage de script, orienté objet et prototype. Il permet d'ajouter de l'interactivité aux animations Flash, en répondant aux actions de l'utilisateur, et en pilotant les Movies clip qui sont des conteneurs graphiques permettant de hiérarchiser les animations, et les différents objets multimédias comme les images, le son, la vidéo, Il permet également la communication de l'application avec le serveur, notamment par le chargement de fichiers ou la communication avec un langage serveur comme le PHP.

VISUAL BASIC (.NET)

C'est un langage de programmation événementielle créée par Microsoft en 1991, il est un dérivé du Basic. Le Visual Basic permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôles ou objets ActiveX. Il a été conçu pour s'apprendre facilement. Programmer en VB est un mélange de plusieurs tâches, comme disposer visuellement les composants et les contrôles sur les formulaires, définir les propriétés et les actions associées à ces composants, et enfin ajouter du code pour ajouter des fonctionnalités. La facilité de mise en œuvre d'un projet fait que beaucoup de projets Visual Basic sont à refaire entièrement car ils ont été mal développés par des débutants Visual Basic. Ce langage possède une évolution : Visual Basic .Net qui possède énormément d'évolutions tels que l'orientation objet, la prise en compte du framework .Net.

OPENEDGE ADVANCED BUSINESS LANGUAGE (ABL)

C'est un langage impératif créé en 1984 par Progress Software Corporation (PSC) qui a changé le nom du langage en 2006 qui s'appelait précédemment Progress ABL. Il est classé comme un langage de programmation de quatrième génération. Il permet de faire des requêtes directement sur la base de données sans utiliser un outil particulier, il n'utilise pas la syntaxe de SQL pour faire cela.

LES COMPILATEURS

GNU COMPILER COLLECTION (GCC)

GCC est un ensemble de compilateurs créés par le projet GNU. Il est un logiciel libre capable de compiler divers langages de programmation, dont C (GCC), C++ (G++), Objective-C (GobjC), Java (GNU Compiler for Java), Ada (GNU New York University Ada Translator) et Fortran (GNU Fortran). Il possède aussi un débogueur GNU Debugger. Ce compilateur est utilisé pour le développement de la plupart des logiciels libres comme le noyau Linux. C'est Richard Stallman qui débute son développement avec le langage de programmation Pastel, un « dialecte » non portable de Pascal, et dont la première version a été publiée en 1984. Puis grâce à Len Tower, Michael Tiemann et d'autres contributeurs en 1987, GCC supporte plus d'une douzaine de d'architecture. Il est notamment le premier compilateur à supporter le C++. En 1992, la version 2.0 apporte en plus des nombreuses optimisations dont un support stable du langage C++. En 1997, un groupe de développeurs trouve le modèle de développement lent et peu propice aux améliorations, ils décident alors de faire un fork du projet et le nomment EGCS (pour « Experimental/Enhanced GNU Compiler System »). Et en avril 1999, EGCS et GCC sont réunis. Depuis, GCC suit de près l'évolution de la normalisation des langages, et parfois même la précède. Il est le compilateur le plus utilisé dans la communauté des logiciels libres et est le compilateur de nombre de systèmes d'exploitation comme GNU/Linux, les BSD ou encore Mac OS X.



Figure 23 : GNU Compiler Collection (GCC)

MONO

Mono est une plateforme de développement Microsoft .Net open-source, qui a été initié par Miguel de Icaza un employé de Novell. Il comporte un compilateur libre de C# 3.0, une machine virtuelle, de nombreuses classes de base, un IDE : MonoDevelop. Il possède aussi de nombreuses API indépendantes de l'environnement, destinées à la programmation sous GNU/Linux et plus particulièrement GNOME, compatibles avec le framework .Net de Microsoft.

Au niveau des licences, le compilateur C# et les outils de Mono sont distribués sous licence GNU GPL, les bibliothèques d'exécution sont sous licence GNU LGPL et les bibliothèques de classes sous X11. Cependant, il peut être dangereux de travailler avec Mono à cause des violations de brevet de Microsoft, bien qu'il y est eu la signature d'un accord entre la société Novell, qui détient les droit sur Mono, et Microsoft le 2 novembre 2006 où Microsoft s'engage à ne poursuivre ni Novell, ni leurs clients, pour violation de brevets logiciels.



Figure 24 : Mono

VISUAL C++ / C# / BASIC

Visual C++/C# est un environnement de développement intégré pour Windows, conçu par Microsoft pour les langages de programmation C/C++/C#/Basic. Il intègre différents outils pour développer, compiler, déboguer un programme ou des bibliothèques s'exécutant sur Windows. Il a été intégré dans le framework Visual Studio, qui constitue un cadre unique aux divers environnements de développements de Microsoft. Ils sont sous licence Microsoft et ils sont donc propriétaire.



Figure 25 : Microsoft

RUBY MRI

Ruby MRI vient de Matz's Ruby Interpreter créé par le créateur de Ruby : Yukihiro Matsumoto, il correspondait à l'implémentation de référence du langage de programmation Ruby, mais il a été remplacé le 1^{er} janvier 2007 pour la version 1.9 de Ruby par l'interpréteur YARV. Il est sous licence GNU GPL et Ruby.

YET ANOTHER RUBY VM (YARV)

YARV est un interpréteur Ruby écrit par Koichi Sasada qui a remplacé l'interpréteur fait par l'auteur de Ruby car il apporte un gain notable en performances. Cet interpréteur a été réalisé en Ruby et a été publié avec la double licence GNU GPL et Ruby.

JAVA DEVELOPMENT KIT (JDK) - JAVA RUNTIME ENVIRONMENT (JRE) - JAVA VIRTUAL MACHINE (JVM)

La machine virtuelle Java permet d'exécuter des programmes compilés sous forme de bytecode Java. Il y en a une différente pour chaque système : Linux, Windows, Mac OS, IBM, etc... ce qui permet aux applications Java compilées en bytecode de produire les mêmes résultats quelle que soit la plate-forme.

Le Java Runtime Environment contient la JVM, et aussi la bibliothèque standard à partir de laquelle sont souvent développés tous les programmes en Java ce qui garantit la portabilité de l'application.

Le Java Development Kit correspond au JRE ainsi que de nombreux autres outils de développement comme le compilateur (javac) qui transforme le code Java en bytecode pour que la machine virtuelle Java puisse l'interpréter, le générateur de documentation (javadoc), l'archiveur (jar) et plusieurs autres outils. Il existe plusieurs JDK car elles correspondent à différentes plateformes tels que le JDK pour Java 2 Standard Edition (J2SE), le JDK pour Java Enterprise Edition (J2EE), etc...

COMPILATEUR ADOBE : MXMLC

C'est un compilateur libre qui est intégré dans le logiciel propriétaire Adobe Flex. La compilation s'effectue en ligne de commande et nécessite la JRE de Java. Cependant, la société Adobe ne diffuse pas cette information, il faut la trouver sur Internet ou alors utiliser leur logiciel Adobe Flash qui est propriétaire. Je n'ai cependant pas réussi à trouver plus d'information sur ce compilateur.

HAXE

Il correspond à un langage de programmation multiplateforme créé par Nicolas Cannasse et la société Motion-Twin dont il fait partie. Ce langage est sous licence GNU GPL version 2. Il permet de compiler un programme fait avec HaXe en Javascript, NekoVM, PHP, C++ et Flash, prochainement en Java et C#. Il permet aussi la compilation de programme écrit en ActionScript sans passer par le compilateur ou les produits d'Adobe.

COMPILATEUR OPENEDGE ABL

Ce compilateur fait partie d'une collection d'outils qui intègre une base de données relationnel et des modules de développement tels que l'AppBuilder qui permet de faire l'interface d'une fenêtre et d'écrire les programmes, le Data Dictionary qui permet de gérer la base de données. La société Progress fournit une version permettant d'exécuter les programmes compilés, cette version est multiplateformes : Unix, Linux et Windows.

CONCLUSION

On remarque, dans le monde de l'informatique, que tous les compilateurs ne sont pas soumis aux mêmes licences et donc aux mêmes engagements. Avant d'utiliser un logiciel ou un compilateur propriétaire pour développer nos applications, on peut rechercher s'il n'y a pas un équivalent dans le monde du libre. Si ce n'est pas le cas regarder si un autre langage ne serait pas plus approprié pour notre futur application, car si on utilise quelque chose de propriétaire on est dépendant de lui et donc limité. Si le propriétaire du logiciel décide d'abandonner une partie des fonctionnalités que l'on utilise, on n'aura pas d'autre choix que d'accepter et essayer de trouver une solution externe pour retrouver ces fonctionnalités.

CONCLUSION

Durant cette deuxième année en entreprise, j'ai appris à utiliser Progress Software au sein d'un projet web et à y intégrer mes connaissances en matière de développement web pour le rendre compatible W3C, mais aussi à générer un fichier EDI avec un format particulier de données. De plus, cette année j'ai appris à mettre en place l'outil Redmine sur une plateforme Windows Server et Ubuntu Server.

Cependant, le gros point fort de cette alternance se trouve au niveau du développement d'une application en équipe. En effet, c'est le premier stage où j'ai pu intégrer une équipe de développement et où j'ai pu apprécier le travail d'équipe dans une bonne ambiance.

Pour une bonne insertion au sein du Groupe Cogeser, il faut aimer travailler en équipe et dans la bonne humeur car vous aurez à un moment ou un autre besoin de quelqu'un pour avancer dans votre projet (car il y a très peu de documentation sur Progress) et le mieux pour travailler dans de bonne condition, c'est la bonne entente au sein d'une équipe.

NOTES

- (1) FSF : La Free Software Foundation est une organisation à but non lucratif fondée par Richard Stallman le 4 octobre 1985. Elle assure la promotion et la défense des logiciels libres au niveau international. Elle est à l'origine de la General Public Licence (GPL).
- (2) Rétro-ingénierie : peut-être aussi appelé rétroconception, ingénierie inversée ou ingénierie inverse, cela consiste à étudier un objet pour en déterminer le fonctionnement interne ou la méthode de fabrication.
- (3) Paradigme : est le style de programmation qui traite de la manière dont les solutions aux problèmes doivent être formulées, par exemple : programmation impérative, orientée objet, déclarative, fonctionnelle, récursive, etc...

TABLES DES ILLUSTRATIONS

<i>Figure 1 : Répartition des sociétés de Cogeser.....</i>	<i>10</i>
<i>Figure 2 : Organigramme de la société</i>	<i>11</i>
<i>Figure 3 : Exemple de fichier EDI.....</i>	<i>14</i>
<i>Figure 4 : Redmine</i>	<i>16</i>
<i>Figure 5 : WEBN2F</i>	<i>17</i>
<i>Figure 6 : Le logiciel libre</i>	<i>20</i>
<i>Figure 7 : Langage C/C++</i>	<i>21</i>
<i>Figure 8 : Langage Java.....</i>	<i>21</i>
<i>Figure 9 : Langage Ruby.....</i>	<i>21</i>
<i>Figure 10 : Le logiciel propriétaire.....</i>	<i>22</i>
<i>Figure 11 : Visual Basic (Microsoft).....</i>	<i>23</i>
<i>Figure 12 : ActionScript (Adobe).....</i>	<i>23</i>
<i>Figure 13 : Advanced Business Language (Progress Software).....</i>	<i>23</i>
<i>Figure 14 : GNU General Public License.....</i>	<i>25</i>
<i>Figure 15 : GNU Lesser General Public License</i>	<i>26</i>
<i>Figure 16 : GNU Affero General Public License</i>	<i>26</i>
<i>Figure 17 : GNU All-Permissive License</i>	<i>26</i>
<i>Figure 18 : Licence CCO.....</i>	<i>26</i>
<i>Figure 19 : Licence CeCILL.....</i>	<i>27</i>
<i>Figure 20 : Licence Ruby.....</i>	<i>28</i>
<i>Figure 21 : Licence X11.....</i>	<i>29</i>
<i>Figure 22 : JSON license.....</i>	<i>30</i>
<i>Figure 23 : GNU Compiler Collection (GCC)</i>	<i>33</i>
<i>Figure 24 : Mono.....</i>	<i>34</i>
<i>Figure 25 : Microsoft.....</i>	<i>34</i>

SOURCES DOCUMENTAIRES

<http://fr.wikipedia.org/>

<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

<http://www.gnu.org/>

<http://tel.archives-ouvertes.fr/docs/00/46/78/49/PDF/Benkeltoum.2009.TheseOpenSource.pdf>

<http://www.halpanet.org/?q=node/11>

http://www.gnu.org/prep/maintain/html_node/License-Notices-for-Other-Files.html

<http://creativecommons.org/publicdomain/zero/1.0/legalcode>

<http://www.cecill.info/licences.fr.html>

<http://www.ruby-lang.org/en/LICENSE.txt>

<http://www.xfree86.org/3.3.6/COPYRIGHT2.html#3>

<http://hub.opensolaris.org/bin/download/Main/licensing/cddllicense.txt>

<http://www.json.org/license.html>

<http://www.ruby-lang.org>

<http://www.java.com/fr/>

<http://haxe.org/>