

Ministère de l'Éducation Nationale, de la Recherche et de la Technologie

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

INSTITUT d'INFORMATIQUE d'ENTREPRISE

RAPPORT DE STAGE

**Migration de l'environnement de
développement d'enseignants produisant
des logiciels pédagogiques libres**

Florian BOUCAULT
stage de 2^{ème} année

Maître de stage : M. Olivier CORTÈS

Stage effectué à RyXéo SARL
Immeuble Topaze, Entrée C, 2 rue Jean Bonardel,
33140 Villenave d'Ornon

du 30/05/05 au 12/08/05

Remerciements

Je tiens à remercier mes maîtres de stage Eric Seigne et Olivier Cortès pour l'accueil chaleureux qu'ils m'ont réservé. Leur générosité et leur gentillesse ont rendu ces trois mois inoubliables tant humainement que techniquement.

Je remercie également l'administration de l'IIE ainsi que ses professeurs, pour me permettre chaque année de poursuivre mes études d'ingénieur, et ce dans une entente agréable.

Je félicite de tout coeur l'équipe du terrier d'AbulÈdu qui jour après jour contribue à l'introduction de l'informatique libre dans l'éducation.

Enfin, merci infiniment à Célia pour son soutien inconditionnel, qui, je l'espère, durera toute notre vie.

Table des matières

1	Contexte du stage	3
1.1	Les logiciels libres	4
1.1.1	Origine	4
1.1.2	Définitions	4
1.1.3	Les enjeux	5
1.1.4	Etat actuel	7
1.2	« RyXéo : une fenêtre ouverte sur l’informatique libre »	8
1.2.1	Histoire et philosophie	8
1.2.2	Activités de l’entreprise	8
1.3	Objectif du stage	10
1.3.1	But pédagogique	10
1.3.2	Point de départ	10
1.3.3	Problématique abordée	10
2	Un nouvel environnement de développement pour le terrier d’AbulÉdu	12
2.1	Cahier des charges	13
2.2	Etat de l’art et sélection	13
2.2.1	Langage de programmation	13
2.2.2	Bibliothèque graphique	14
2.2.3	Interface de développement	15
2.2.4	Outil d’aide au dessin d’interfaces graphiques	16
2.3	Transmission des connaissances	16
2.3.1	Réalisation d’un exemple et d’un début de bibliothèque	17
2.3.2	Ecriture de documentation	17
2.3.3	Séances d’initiation	17
2.3.4	Accompagnement technique	18
3	Bilan du travail effectué	19
3.1	Améliorations possibles	20
3.2	Importance du travail d’analyse	20
3.3	Apport du stage	20
	Références	21
	Annexes	23

Chapitre 1

Contexte du stage

1.1 Les logiciels libres

1.1.1 Origine

Durant les années 1960 et 1970, le logiciel était fréquemment partagé librement entre les encore peu nombreux utilisateurs d'ordinateurs, lesquels, à l'époque, étaient principalement de grands serveurs centraux. À la fin des années 1970, les entreprises impliquées commencèrent à imposer régulièrement des restrictions d'utilisation de leurs logiciels au travers des premières licences logicielles.

En réponse à cette fermeture de ces libres échanges qui avaient lieu auparavant entre développeurs, Richard Stallman, alors chercheur au laboratoire d'intelligence artificielle du MIT (Massachusetts Institute of Technology), remit en cause cette évolution qu'il jugeait contraire aux principes de collaboration communément admis dans la recherche scientifique. Il posa alors la définition originale des logiciels libres, informellement d'abord (1985), puis plus tard juridiquement (1989) au travers de la licence publique générale (GPL) [8]. Il mis en route le projet GNU [14] fin 1983, destiné à fournir un système d'exploitation compatible UNIX (respectant les standards POSIX [11]) entièrement libre pour tous. En 1985, il fonda la Free Software Foundation (FSF) [7], une organisation à but non lucratif destinée à appuyer le mouvement du logiciel libre et plus particulièrement le projet GNU.

Cette fondation aide aujourd'hui juridiquement de nombreuses entreprises et développeurs de logiciels libres. En effet, une très large communauté, composée non seulement d'informaticiens mais aussi de particuliers de tous horizons ainsi que de sociétés de tailles diverses, a fait le choix de suivre cette philosophie de partage et d'entraide.

1.1.2 Définitions

Les logiciels libres garantissent quatre libertés fondamentales à l'utilisateur :

- La liberté d'exécuter le programme quel qu'en soit l'usage,
- La liberté d'étudier le programme,
- La liberté de copier et redistribuer le programme,
- La liberté d'apporter des modifications au programme et de les rendre publiques.

Deux de ces libertés nécessitent l'accès au code source étant donné que l'étude d'une version binaire d'un logiciel est très difficile et assez inefficace comparée à la lecture d'un code source documenté. En aucun cas les logiciels libres ne sont comparables aux gratuits (prix nul), aux partagiciels (fonctionnalités complètes contre un prix) ou aux logiciels du domaine public. Un logiciel libre peut être vendu et il n'est pas forcément le fruit d'un bricolage d'amateurs.

De plus, la volonté de préserver libres lesdits logiciels a amené au concept du *copyleft* où toutes les modifications apportées doivent elles aussi être libres.

D'après le Livret du Libre [18] : "Le copyleft est une utilisation particulière du droit d'auteur partant du principe que le partage doit fonctionner dans les deux sens. Il autorise la copie, la modification et la diffusion d'une oeuvre, en imposant que les versions modifiées faisant l'objet d'une diffusion soient également disponibles sous une licence copyleft. Ainsi, avec le copyleft, ce qui est libre reste libre pour toujours."

Et d'après la définition originale du projet GNU [13] : "Copyleft is a general method for making a program or other work free, and requiring all modified and extended versions of the program to be free as well."

La possibilité donnée par un auteur aux utilisateurs de copier, utiliser, étudier, modifier et distribuer son oeuvre a été étendue à de nombreux types de media, comme la musique [16], les documents écrits [10][15] ou encore toute oeuvre artistique [3] au travers de licences adaptées [1].

1.1.3 Les enjeux

Sociaux

Le libre tente d'enrayer la spirale descendante du verrouillage de l'innovation, des idées, de la connaissance et des moyens de communication.

Faire le choix de placer un logiciel sous une licence libre n'est pas une révocation du droit de propriété intellectuelle puisqu'au contraire il prend ses racines dans la reconnaissance des auteurs et dans l'incitation à la création, sources du droit de la propriété intellectuelle. Selon Jean-Benoît Zimmermann [12], "la question fondamentale de la propriété intellectuelle [est] de concilier incitation à l'innovation et diffusion technologique, [celle-ci] débouche sur un paradoxe, parfois dénommé dilemme shumpeterien, entre performance individuelle et efficacité collective." Ce dilemme n'a pu être satisfait par le dialogue à sens unique instauré par les logiciels propriétaires où n'est pris en compte qu'un seul objectif : la protection des intérêts de l'auteur. Ceci est à l'opposé des logiciels libres comme le souligne Etienne Lavanant dans son essai sur Pourquoi le libre ne concerne pas que les informaticiens [5] : "Le mouvement du libre tente aujourd'hui de faire en sorte que la propriété intellectuelle continue de servir l'intérêt de la communauté plutôt que les intérêts de quelques groupes d'individus."

Par ailleurs, ce partage rejoint celui de la connaissance. Celle-ci s'est vue de plus en plus librement échangée avec la généralisation d'Internet et notamment de l'utilisation des weblogs (journaux électroniques), et du principe du Wiki (où chacun peut prendre part à l'écriture du contenu). L'exemple le plus flagrant de cette évolution induite par le libre est sans doute l'encyclopédie électronique collaborative Wikipédia [2] où en seulement 4 ans le nombre d'articles a atteint 200 000, sans pour autant sacrifier à la qualité.

Enfin, la pérennité des communications à travers l'outil informatique apparaît clairement tributaire non seulement de logiciels libres mais aussi de formats et de langages libres. De nombreux gouvernements l'ont d'ores-et-déjà compris et recommandent ou imposent leur utilisation. Dans les faits, le réseau Internet dans son ensemble repose en majorité sur des ordinateurs fonctionnant sous des systèmes d'exploitations libres ou tout du moins sur des composants libres comme les services de pages hypertextes (Apache, 70% des serveurs en Octobre 2005 d'après Netcraft [17]), de courrier électronique (Sendmail, Exim et Postfix, 41% des serveurs en Avril 2004 d'après Falko Timme [6]) ou encore de noms (Bind). De manière évidente, le réseau nécessite aussi des standards de communications ouverts. Concernant plus particulièrement le *stockage* des informations, seule l'adoption de formats ouverts [9] et normalisés garantit l'indépendance des utilisateurs (particuliers ou non) vis-à-vis de leurs outils [4]. En informatique, les langages de programmation populaires sont souvent libres

(ex. : C, Python, Perl, PHP), c'est-à-dire des langages où chacun peut avoir accès à ses spécifications et à des compilateurs ou interpréteurs. Ceux-ci déverrouillent l'accès à l'interopérabilité en introduisant des outils communs.

Economiques

Les conséquences économiques qui viennent les premières à l'esprit sont souvent celles impactant les sociétés développeuses de logiciels qui se retrouveraient sans revenus du fait de l'apparente gratuité des solutions libres. Cependant, il faut d'abord noter que fréquemment, des logiciels libres sont effectivement vendus tel que recommandé par la Free Software Foundation. D'autre part, comme remarqué dans le Livret du Libre [18] : "Contrairement à une croyance répandue, une majorité d'informaticiens travaille sur des logiciels qui ne sont pas vendus. Ils peuvent être destinés à un usage interne ou bien ils n'existent que pour permettre d'utiliser un autre produit : un service en ligne, un périphérique matériel..." L'emploi dans l'industrie informatique n'est par conséquent pas en mis en péril par le libre, mais en partie remanié puisque le temps passé à redévelopper l'existant n'a plus lieu d'être. Celui-ci se voit remplacé par un travail plus important de réutilisation et d'intégration de composants déjà conçus rendant possible une meilleure adaptation aux besoins réels des utilisateurs.

Cette mutualisation des développements entraîne par là même une diminution des coûts associés et permet de réduire les risques de monopolisation d'un marché par une société unique.

Les coûts d'acquisition des licences des outils mêmes de développement et de gestion sont eux aussi drastiquement rabaisés.

Enfin, un développement libre diminue les risques pour l'entreprise éditrice puisque la connaissance peut être partagée par d'autres personnes ; l'entreprise peut faire appel à des développeurs extérieurs potentiellement plus qualifiés et gagne ainsi en flexibilité.

En termes d'ingénierie

C'est un nouveau modèle de développement qui est apparu avec l'avènement simultané du réseau Internet et des logiciels libres. L'exemple le plus marquant étant celui de la réalisation collaborative du noyau Linux initiée en 1991 par Linus Torvalds. Celui-ci est souvent qualifié de *bazar* puisqu'en apparence il n'y a pas de hiérarchie pyramidale classique. Chacun est à même de participer et l'approbation des contributions se base uniquement sur des critères techniques. Ce changement radical dans la façon de mener un projet est décrit, analysé et expérimenté de manière fine dans l'ouvrage d'Eric S. Raymond, La cathédrale et le bazar [19]. En conséquence de la mise à contribution de tous, la force de travail s'en voit potentiellement accrue. On constate d'ailleurs que la réactivité face à des erreurs de programmation ou de simples questions d'utilisation est souvent meilleure pour des produits libres que pour leurs équivalents propriétaires. De fait, la sécurité en est améliorée ; de surcroît, la découverte de failles est définitivement plus efficace avec un code ouvert largement consulté. Le rapprochement de l'utilisateur et du développeur ramène le travail d'ingénierie à ses fondements en répondant de manière plus adéquate encore à la demande.

Par l'utilisation de technologies libres, les liens imposés entre le fournisseur et l'utilisateur disparaissent. Le fournisseur est donc forcé d'instaurer des relations

saines s'il souhaite conserver sa crédibilité. Cette indépendance est d'ailleurs requise par les gouvernements.

1.1.4 Etat actuel

Les concepts véhiculés par le mouvement du libre ont eu et ont encore bien du mal à percer, que ce soit en entreprise, chez les particuliers ou dans les administrations. On assiste tout de même à de nombreux passages au libre bien souvent pour des raisons économiques mais aussi pour garantir l'indépendance technologique et la pérennité des informations.

Des rassemblements de tailles variables sont régulièrement organisés à travers le monde et de très riches ressources libres sont maintenues par une grande quantité de sites Internet tant professionnels que personnels.

Les licences libres quant à elles sont de mieux en mieux reconnues par les diverses entités juridiques et la jurisprudence s'applique chaque jour un peu plus que ce soit pour défendre les développeurs ou les utilisateurs.

Des preuves de la viabilité du modèle économique du libre ont été apportés par de multiples entreprises, petites ou plus importantes.

Cependant, de fortes menaces pèsent constamment sur l'avenir du libre. Des mesures telles que le projet de loi DADVSI visant à contrôler le système de l'utilisateur imposent le secret à l'opposé de l'ouverture et de la transparence caractérisant les logiciels libres. L'acceptation potentielle des brevets logiciels en Europe rendrait illégaux de très nombreux logiciels libres déjà existants et marginaliserait les petites structures au profit des géants, seuls à même d'effectuer les complexes recherches d'antériorité et de payer le prix des brevets.

1.2 « RyXéo : une fenêtre ouverte sur l’informatique libre »

1.2.1 Histoire et philosophie

RyXéo est une Société de Services en Logiciels Libres (SSLL) née à l’automne 2003 lorsque M. Olivier Cortès rejoignit la société individuelle Rycks fondée par Eric Seigne trois années plus tôt. Le coeur de métier de l’entreprise est la recherche et le développement. Elle produit plus particulièrement la solution logicielle libre AbulÉdu Professionnel, destinée à faciliter la création et l’usage d’un réseau informatique dans un établissement scolaire.

La société contribue au tissu économique et social tant dans la région que dans le pays et dans le monde entier :

- Localement, de par son appartenance au réseau ProLibre, une section de l’ABUL rassemblant les entreprises d’Aquitaine qui travaillent dans le libre.
- D’un point de vue national : à travers les dépoyeurs, les distributeurs et les partenaires AbulÉdu Professionnel qui se répartissent dans plus de vingt départements en France.
- Au niveau international, RyXéo intervient en Afrique dans le cadre de formations ou de conférences pour le développement durable et l’indépendance technologique de l’Afrique.

Le libre est la principale préoccupation de l’entreprise : il intervient dans les développements durables, une des clés de l’avenir de nos nations. RyXéo ne peut être définie comme une SSII qui fait du libre, mais plutôt comme une entreprise qui adopte la philosophie du libre comme une façon d’exister. Et ceci, en plus des développements logiciels, rend la gestion d’entreprise et des ressources plus passionnantes.

Voici comment RyXéo qualifie le libre :

“Le libre est une façon plus équitable de voir l’informatique. Les fournisseurs d’un logiciel ne peuvent plus enchaîner leurs clients en ne leur donnant pas les sources de leurs programmes, ou en restreignant le logiciel à telle ou telle utilisation. Ils leur permettent d’être libre de choisir leur prestataire. Ceci est appréciable en cas de coup dur (cession d’activité du prestataire), et permet de construire entre le prestataire et son client une relation de confiance plutôt qu’une relation dominant/dominé.” [20]

1.2.2 Activités de l’entreprise

Le développement et le conseil

Jour après jour, RyXéo conçoit, programme et adapte des logiciels libres pour qu’ils répondent au mieux aux besoins de ses clients ainsi qu’aux siens. RyXéo est à la pointe des technologies libres et OpenSource, est partie prenante de nombreux projets, et avance *avec* les logiciels libres. Fort de son panel de compétences vaste et solide, RyXéo est à même de conseiller et d’assister ses clients au fur et à mesure de leur évolution technologique.

La conception, l'installation et la maintenance de réseaux

La mise en place d'infrastructures réseaux sécurisées de qualité pour les PME/PMI et pour les établissements scolaires à base de logiciels libres est la tâche principale de l'entreprise. Pare-feux, serveurs mandataires, serveurs de messagerie, d'applications, de partage de ressources sont autant de composants que l'équipe maîtrise très finement. La prestation peut aller de la simple assistance à la gérance externalisée selon les désirs et les besoins de chaque structure.

La technologie maîtresse de RyXéo est la solution libre AbulÉdu Professionnel. Elle permet la mise en place d'un réseau pédagogique performant, fiable et modulaire. Particulièrement adapté pour l'enseignement, cette solution convient également aux PME souhaitant s'équiper d'un réseau de bureautique simple et efficace. L'accent est mis sur la facilité d'administration sans pour autant sacrifier à la sécurité. La composante logicielle peut-être accompagnée de tout type de matériel que RyXéo certifie puis déploie. Actuellement, plus de 250 serveurs AbulÉdu sont en activité et surveillés en permanence par l'équipe.

La formation

RyXéo intervient pour des organismes de formations agréés sur les logiciels libres ou des sujets connexes. Les formations sont sur mesure : les supports et le contenu sont adaptés aux besoins ; l'étude de cas concret et les exercices en situation sont les principes directeurs de ces cours.

Dans la démarche de qualité AbulÉdu Professionnel, sont organisées des formations de différents niveaux pour ses revendeurs précédant leur certification.

RyXéo participe également à de nombreux colloques nationaux et internationaux comme les «Linux Expo», les Rencontres Mondiales du Logiciel Libre, ou les Rencontres Africaines du Logiciel Libre, dans le cadre de développements durables et pour participer à l'émancipation technologique des pays d'Afrique Francophone.

La vente de matériel

De manière plus anecdotique, RyXéo est revendeur de matériel informatique pour les particuliers au travers de sa boutique en ligne (boutique.ryxeo.com). Bien entendu, un point d'honneur est mis sur la mise à disponibilité de matériel pour lesquels des pilotes libres existent.

1.3 Objectif du stage

1.3.1 But pédagogique

En tant que stage de seconde année de l'Institut d'Informatique d'Entreprise, les objectifs sont multiples :

- mettre en application les techniques informatiques enseignées à l'école,
- développer et démontrer des compétences d'analyse et de conception,
- faire preuve d'une bonne capacité d'intégration à l'environnement de travail et notamment à l'équipe d'accueil,
- et répondre efficacement aux attentes de l'entreprise.

Dans le contexte d'une SSLL telle que RyXéo où tous les moyens nécessaires à l'accomplissement de mon travail ont été mis à ma disposition, la satisfaction de ces objectifs m'était grandement facilitée.

1.3.2 Point de départ

Le terrier d'AbulÉdu est une suite d'applications libres développée par des enseignants. Avec l'aide technique d'Eric Seigne, ils se sont mis à développer des outils adaptés à *leurs* besoins pédagogiques spécifiques. Ceux-ci sont inclus dans AbulÉdu mais sont bien souvent développés de manière à en être indépendants, c'est à dire qu'ils sont pour la plupart utilisables sur n'importe quel système GNU/Linux, MacOS ou Windows. Ce sont à présent près d'une quinzaine de logiciels ludiques, ludo-éducatifs et pédagogiques développés par des non informaticiens qui sont utilisés quotidiennement par de nombreuses écoles et parents d'élèves.

Depuis le commencement du terrier en 2002, le langage Tcl, associé à sa boîte à outils graphiques naturelle Tk, a été employé pour la réalisation de celui-ci. Sans phase de compilation et bénéficiant d'une large communauté et d'une importante stabilité, c'était un choix judicieux d'outil de travail pour des enseignants béotiens en matière de programmation informatique.

Cependant, l'expérience faisant, quelques difficultés virent le jour. Du côté graphique, une mauvaise intégration à l'environnement de bureau et une apparence un peu désuète du Tk semblaient de plus en plus réhhibitoires. De plus, l'internationalisation des textes dépendants du système d'exploitation n'était pas totalement satisfaisante et de nouveaux langages plus simples en termes de syntaxe et de maintenance étaient apparus.

1.3.3 Problématique abordée

Les développeurs des logiciels du terrier sont majoritairement des professeurs qui n'ont a priori aucune formation en informatique. Ils ont développé ces outils pédagogiques à l'aide de Tcl/Tk mais la nécessité d'une autre voie de programmation s'est faite ressentir avec des attentes assez précises de la part des enseignants. Il m'a donc fallu commencer par m'imprégner simultanément des environnements de l'éducation et du libre afin d'épouser au mieux la demande. Ce fut un travail continu, qui après de nombreuses discussions, lectures, rencontres, visites et tests s'est révélé extrêmement gratifiant et révélateur. De plus, le temps, la patience et les ressources des personnes impliquées n'étant pas

illimités, le choix de ces outils se devait d'être, d'une part, au mieux de mes possibilités d'analyse, et d'autre part, correctement introduit et justifié. S'ensuivit donc une période de recherche aussi exhaustive que possible dans la mesure du relativement court temps imparti. Une fois les décisions prises en accord avec les dialogues permanents, il s'agissait de transmettre la nouvelle ainsi que les connaissances requises pour commencer à se mettre à l'ouvrage dans ce nouvel environnement de développement. Enfin, une période de support utilisateur et d'accompagnement technique s'enchaîna naturellement.

Chapitre 2

Un nouvel environnement de développement pour le terrier d'AbulÉdu

2.1 Cahier des charges

Il s'agissait de trouver une nouvelle plateforme de développement de logiciels graphiques répondant aux exigences principales suivantes :

- Destinée aux enseignants donc **simple** à apprendre et à utiliser
- **Portable** sur GNU/Linux, Windows et MacOS/X avec un minimum d'effort
- Ayant un **rendu natif** à l'interface graphique du système d'exploitation
- Tout en restant assez **léger** pour l'ordinateur
- **Libre** et reposant sur des **outils libres**
- Capable de gérer des entrées et des sorties **audio**
- Facilitant l'**internationalisation** des logiciels développés

D'autre part, la version associative d'AbulEdu initialement créée par Eric Seigne repose sur la distribution GNU/Linux Mandrake 7.2 depuis lors et reste encore utilisée par de nombreuses écoles non désireuses de passer à la version professionnelle basée sur la Debian Woody, version stable de cette distribution jusqu'à récemment. Idéalement, le langage et la bibliothèque graphique choisis doivent donc fonctionner sous Mandrake 7.2.

D'autres demandes anecdotiques telles que la possibilité de redimensionner rapidement des images et générer des fichiers au format SVG simplement sont apparues au fur et à mesure de mes recherches.

2.2 Etat de l'art et sélection

Une plateforme telle que celle requise est composée de trois volets : le langage de programmation, la bibliothèque graphique et l'interface de développement. Il s'est rapidement avéré que de nombreuses solutions libres existaient pour chacune des trois composantes. L'enquête a été essentiellement basée sur les résultats de moteurs de recherche Internet en partant des exigences ainsi que de mes propres connaissances et expériences passées. Des critères comme la popularité, synonyme de communauté active, l'ancienneté, ou encore le nombre de structures privées soutenant le projet ont aussi été pris en compte afin de garantir la pérennité des choix.

Dans un souci de rigueur, il était nécessaire d'aller au delà des idées communément admises et de se plonger soi-même un minimum dans l'utilisation de chacune des possibilités. Ainsi, de brèves mais fréquentes expérimentations ont été tentées tout au long des recherches.

2.2.1 Langage de programmation

Tout d'abord, il m'a fallu référencer l'ensemble des langages suffisamment abordables pour dépasser le stade du rejet que l'on peut ressentir au début de l'apprentissage. La famille des langages disposant d'interpréteurs s'est vite imposée puisqu'elle retire des éléments parasites (compilation) et réduit la phase de développement à sa plus simple expression : l'écriture du code source. Cependant, les logiciels du terrier étant destinés à être étendus, maintenus et améliorés sans limite de temps et par de nombreux contributeurs, un langage disposant

d'une syntaxe forçant la lisibilité et d'une bonne robustesse s'est avéré indispensable.

Voici les langages parmi lesquels le choix a été fait :

Caml Sa principale implémentation, l'Objective Caml, est libre, dispose d'une extension orientée objet, de compilateurs et d'interpréteurs sous une kyrielle de systèmes d'exploitations et d'architectures ; développé par l'INRIA depuis 1985, il est fréquemment utilisé en France dans l'apprentissage de la programmation et a vu son utilisation largement diversifiée.

Python Langage interprété à syntaxe positionnelle dont le développement par son auteur, Guido van Rossum, a débuté en 1990 ; une très large communauté, de très nombreuses bibliothèques et une facilité d'écriture et de lecture le caractérisent.

Perl Depuis 1987, Perl fournit à de très nombreux développeurs un langage proche en fonctionnalités au C et à des langages de scripts tels que sed, awk et shell ; cependant, une certaine réputation d'obscurité entoure les codes sources des programmes écrits en Perl.

Common Lisp Doté d'une longue histoire (héritier direct des dialectes Lisp des années 1950), son paradigme reste fondamentalement fonctionnel bien qu'il accepte des techniques impératives et orientées objets ; il est aujourd'hui bien moins répandu qu'auparavant.

Scheme Dérivé du Lisp dans le but de l'épurer et le simplifier sans pour autant sacrifier à sa puissance expressive.

LOGO Langage de programmation adapté du Lisp dans les années 1960 au MIT qui sert essentiellement à l'introduction à la programmation.

Basic et dérivés Langage inventé en 1964 par John George Kemeny et Thomas Eugene Kurtz qui fut conçu de façon à permettre aux étudiants des filières non scientifiques d'utiliser les ordinateurs.

Il est vite apparu que le Basic et le LOGO étaient trop simplistes et pas assez structurant pour le besoin bien que des interfaces de développements totalement intégrées telles que Gambas ou HBasic existaient. L'approche fonctionnelle du Common Lisp, du Scheme aurait sans doute dérouté les enseignants habitués au Tcl. Le Caml, très expressif et performant, semblait ne pas compter assez de ressources, notamment concernant les bibliothèques disponibles et la taille de sa communauté.

Il restait donc le Python et le Perl, deux langages extrêmement répandus et populaires. Ce sont ici les impératifs en termes de clarté de code qui ont prédominés en faveur du Python.

2.2.2 Bibliothèque graphique

Une bibliothèque graphique fournit un ensemble de fonctionnalités permettant la conception d'interfaces graphiques. Les bibliothèques graphiques servent d'interface entre l'affichage (pilote vidéo) et les entrées (clavier, souris...).

Les bibliothèques bas niveau (Xlib pour les serveurs X) sont assez pénibles à utiliser pour créer une application complète. C'est pour cela qu'ont été créés des bibliothèques de plus haut niveau contenant des fonctions pour créer des fenêtres, des boutons, des menus plus simplement.

Souvent, ces bibliothèques graphiques intègrent aussi toute une panoplie de fonctions pour traiter le réseau, le multimédia, le multithreading, etc... Voici les majeures du monde des logiciels libres qui soient multiplateformes et utilisables en Python :

- Qt** Son développement a débuté en 1991. Au départ propriétaire, elle est passée sous une double licence non libre pour le système Microsoft Windows. Finalement, en juillet 2005, à la sortie de sa version 4, la bibliothèque Qt a été rendue sous licence GPL pour tout projet libre. Elle est plébiscitée par plusieurs milliers d'entreprises.
- GTK (Gimp ToolKit)** Originellement créée pour les besoins du logiciel de traitement d'images "The GIMP", elle est à présent utilisée pour le développement d'environnements de bureau (GNOME, Xfce, ROX...) et de nombreuses autres applications.
- Tk** Introduite au départ comme une extension du langage de script Tcl, il est à présent possible d'accéder à cette bibliothèque au travers de nombreux langages tels que Perl, Python ou encore Ruby. Il lui manque malheureusement de nombreux composants modernes.
- FLTK** Destinée à être avant tout très légère d'un point de vue mémoire et temps de calcul, sa communauté d'utilisateurs est à l'image du nombre de composants disponibles : restreinte.
- FOX** Bibliothèque assez récente (1997) qui a le défaut majeur de ne pas fonctionner sous MacOS/X.
- wxWidgets** Mise en production en 1992, elle se différencie par l'usage des bibliothèques natives aux plateformes ce qui lui concède une apparence bien intégrée et une bonne rapidité d'exécution.

La bibliothèque FOX, ne fonctionnant pas sous MacOS/X et présentant une apparence constante quel que soit l'environnement, s'est rapidement vue écartée.

Les bibliothèques FLTK et Tk se sont avérées être trop limitées pour les besoins des logiciels du terrier.

Restaient donc GTK, Qt et wxWidgets, tous trois respectant avec plus ou moins d'efforts le cahier des charges établi auparavant.

GTK, très populaire restait cependant assez peu concluant du point de vue de son adaptation visuelle à l'environnement.

Au moment de ces recherches, la bibliothèque Qt, bien que permettant une intégration visuelle naturelle (fonctionnalité d'ailleurs assez récente), posait problème concernant sa double licence d'utilisation.

wxWidgets présentait comme avantage majeur d'être une méta-bibliothèque lui permettant de prendre systématiquement l'apparence de son environnement. De plus, elle intègre la gestion du son, de l'internationalisation et possède si besoin est des implémentations en de très nombreux langages (Python, Perl, BASIC, Lua, Eiffel, JavaScript, Java, Ruby, Haskell, .Net...).

2.2.3 Interface de développement

Deux types d'outils peuvent être utiles à l'utilisation du Python : un éditeur de texte avancé reconnaissant la syntaxe de Python ou une interface de développement

plus ou moins dédiée au Python. Bien que ce choix soit avant tout une affaire de goût, je me suis efforcé de le faciliter aux enseignants en les conseillant et en établissant une sélection. Les critères principaux étaient là encore la liberté des logiciels et leur portabilité.

eric3 Environnement dédié au Python incluant de nombreuses fonctionnalités utiles : débogueur, diagrammes de classes, interface avec serveur de sources... Il semble très bien suivi par l'équipe de développement et plusieurs sociétés l'utilisent comme outil principal de programmation Python.

SPE (Stani's Python Editor) Tout à fait comparable à eric3, il en reprend une bonne partie des possibilités et présente des facilités d'intégration avec wx.

Boa Constructor Bien que disposant de capacités intéressantes, son interface utilisateur s'est révélée très déroutante et difficile à prendre en main.

DrPython Simple et efficace, son système de plugins ouvre une voie facile à l'ajout de nouvelles fonctions.

Eclipse avec le module pydev Une approche très différente à laquelle on peut reprocher sa lourdeur tant visuelle qu'en termes de ressources.

La séduisante simplicité de DrPython en a fait le choix préféré des enseignants.

2.2.4 Outil d'aide au dessin d'interfaces graphiques

L'écriture du code correspondant aux composants graphiques des programmes du terrier est parfois rébarbative et trop technique. Il est souvent pratique de déléguer cette rédaction à des logiciels spécialisés qui, à partir d'une description visuelle de l'interface voulue, s'occupent de produire le code source. Cette approche est souvent utilisée pour de petits projets où le temps disponible est court. Techniquement, ce n'est pas très pratique car le code résultant, bien que pouvant être retouché, ne peut être régénéré après modification sous peine de perdre toutes les corrections manuelles. Ces outils de dessin d'interfaces graphiques privilégient maintenant la transformation vers un format de description d'interface. Le fichier résultant peut ensuite être chargé par un programme Python qui s'occupe d'associer les événements des composants à des actions.

Pour la bibliothèque wxWidgets, trois outils libres existent : wxGlade, wxHatch, XRCed. Les divers essais d'utilisation n'ont malheureusement pas été convainquants. L'emploi d'un outil de design d'interface graphique pour le terrier représentait en fait plus une perte d'énergie qu'un gain de temps.

2.3 Transmission des connaissances

Une fois l'étude des différentes technologies possibles terminée et le choix fait, il s'agissait alors d'apprendre à les utiliser puis de transmettre ces connaissances aux enseignants. Ce passage de savoir s'est fait au travers de documentations, d'exemples mais aussi de séances de cours et d'exercices informels notamment durant les Rencontres Mondiales du Logiciel Libre qui se sont tenues à Dijon.

2.3.1 Réalisation d'un exemple et d'un début de bibliothèque

Le terrier d'AbulEdu comptait parmi ses logiciels celui fort utile pour l'apprentissage du maniement de la souris : le Mulot. Ecrit comme ses comparses en Tcl/Tk, une réécriture en Python/wxWidgets, au moins partielle, s'imposait afin de donner un exemple concret directement lié au terrier.

Cette preuve de concept sert non seulement de base pédagogique mais permet aussi de débiter l'écriture d'un ensemble de modules réutilisables pour l'écriture d'autres logiciels du terrier.

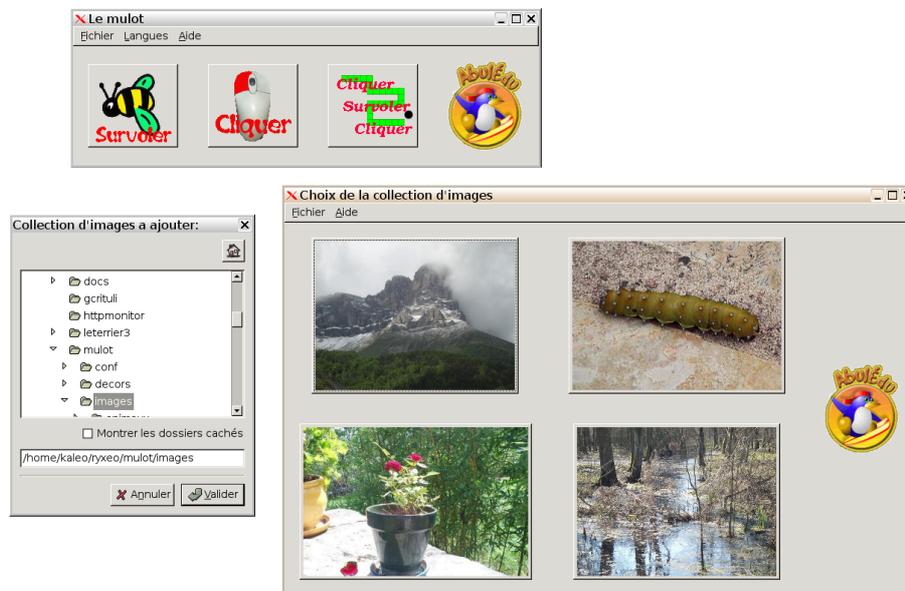


FIG. Captures d'écran du Mulot en wxPython

Le code source correspondant est disponible en annexe.

2.3.2 Ecriture de documentation

Afin d'introduire le Python/wxWidgets aux néophytes, j'ai pu rédiger des documents (attachés en annexe) de découverte et d'initiation. Etant donné qu'une abondante littérature de qualité, et tant en anglais qu'en français, était déjà à disposition de tout un chacun, ces documents sont brefs et font principalement référence à des morceaux choisis d'autres cours. Ils décrivent aussi la mise en place suffisamment détaillée d'un environnement de travail fonctionnel.

2.3.3 Séances d'initiation

A plusieurs reprises, j'ai eu l'occasion de rencontrer en personne les gens impliqués dans le développement des logiciels du terrier. Ceci ne fut pas évident puisqu'il ne faut pas perdre de vue que ces personnes ont créé ces programmes sur leur temps libre. Un des rendez-vous majeurs fut les Rencontres Mondiales du Logiciel Libre 2005 qui ont eu lieu à Dijon. Ces quelques jours permettent aux acteurs des logiciels libres de se rencontrer, et aussi de présenter tant au grand

public qu'aux professionnels intéressés ce qu'est le monde du libre. AbulEdu y avait à sa disposition un stand pour faire des démonstrations, notamment des logiciels du terrier.

Des cours informels sur le Python et la bibliothèque wxWidgets ont donc pu y avoir lieu. Ceci a nécessité une certaine préparation avec entre autres la rédaction d'exercices pédagogiques (attachés en annexe).

2.3.4 Accompagnement technique

Au jour le jour, de nombreuses requêtes émergèrent : des retours sur les documents, exercices, prototypes mais aussi de nouvelles idées de logiciels pédagogiques. Une bonne disponibilité était donc impérative pour des attentes qu'il n'était pas toujours évident de satisfaire.

Chapitre 3

Bilan du travail effectué

3.1 Améliorations possibles

Bien qu'au final la transition du Tcl/Tk vers Python/wxWidgets fut correctement initiée et communément acceptée, des efforts de développement additionnels auraient pu éventuellement déboucher sur un logiciel du terrier entièrement nouveau. Un début d'analyse du projet a vu le jour en fin de stage mais le manque de temps n'a pas permis d'aboutir.

Il aurait été utile de donner des présentations du projet à des développeurs intéressés et disponibles afin de potentiellement recruter, et ainsi ne pas laisser l'équipe du terrier sans expertise technique.

La pédagogie étant un domaine relativement nouveau me concernant, j'ai eu quelques saines difficultés à faire passer mes connaissances informatiques à un public fort heureusement courageux.

Le prototype du Mulot en wxPython était un succès qui a confirmé la qualité de l'analyse préalable. La phase de conception du logiciel a porté ses fruits ; les tests multi-plateformes ont répondu aux attentes du cahier des charges. Cependant, on peut regretter qu'il ne soit pas tout à fait utilisable dans les écoles puisque quelques fonctionnalités du Mulot Tcl/Tk n'ont pas été implémentées.

3.2 Importance du travail d'analyse

L'analyse ici ne concernait pas seulement l'aspect logiciel. En effet, il était crucial de correctement comprendre la philosophie de l'entreprise et de l'équipe du terrier pour bien cerner ce qu'ils attendaient de mon travail. Par exemple, de fréquents questionnements sur le pourquoi des logiciels libres ont modelé ma réflexion de manière à m'adapter à l'environnement de travail. C'est pour cette raison qu'une partie du présent rapport dépeint ce milieu avec une certaine précision.

3.3 Apport du stage

Durant ces douze semaines dans la société, j'ai eu la chance d'aborder le sujet très intéressant des logiciels libres, en particulier dans le milieu de l'éducation, et l'opportunité de tenter de transmettre du savoir informatique.

RyXéo m'a permis d'accroître mes connaissances informatiques dans divers domaines. J'ai pu les approfondir dans des langages tels que le Python, le script shell ou encore le XML ainsi que perfectionner mes compétences en modélisation objet. J'ai également pu travailler mes capacités pédagogiques ce qui aura sans doute été la partie la plus enrichissante de ces trois mois. J'espère avoir apporté une petite pierre à l'édifice de la communauté du libre, édifice qui dépend entièrement de la volonté de chacun.

Enfin, j'ai pu appréhender à nouveau l'importance capitale de la communication et des relations humaines en entreprise.

Bibliographie

- [1] Les licences Creative Commons. <http://creativecommons.org/>.
- [2] Wikipédia, l'encyclopédie gratuite et libre. Internet. <http://wikipedia.org/>.
- [3] La licence Art Libre. <http://artlibre.org/>.
- [4] BASTIEN GUERRY et DARIO TARABORELLI : Pourquoi utiliser des formats ouverts? Internet, 2004. <http://www.openformats.org/fr>.
- [5] ETIENNE LAVANANT : Pourquoi le libre ne concerne pas que les informaticiens. Internet, 14 janvier 2004. http://www.minet.net/spip/article.php3?id_article=110.
- [6] FALKO TIMME : Les statistiques de serveurs mails de Falko Timme. Internet. http://www.falkotimme.com/projects/survey_smtp.php?id=170.
- [7] FREE SOFTWARE FOUNDATION : Le site Internet officiel de la FSF. <http://www.fsf.org>.
- [8] FREE SOFTWARE FOUNDATION : Le texte de la licence GNU General Public License (GPL). Le texte officiel n'est disponible qu'en anglais, pour des raisons légales. Des traductions officielles sont en cours d'études par des juristes. On trouvera une version traduite, non officielle sur le site Internet de la FSF France. <http://www.fsf.org/licenses/gpl.html>.
- [9] GOUVERNEMENT FRANÇAIS : LOI n° 2004-575 du 21 juin 2004 pour la confiance dans l'économie numérique. Journal Officiel n°143, 22 juin 2004. <http://www.legifrance.gouv.fr/WAspad/UnTexteDeJorf?numjo=ECOX0200175L>.
- [10] Le projet Gutenberg. <http://www.gutenberg.org/>.
- [11] IEEE STANDARDS ASSOCIATION : Page de référence de la norme POSIX. <http://standards.ieee.org/regauth/posix/>.
- [12] J.-B. ZIMMERMANN : Logiciel et propriété intellectuelle : du Copyright au Copyleft. *Groupement de Recherche en Economie Quantitative d'Aix-Marseille*, 1999.
- [13] LE PROJET GNU : La définition du copyleft par le projet GNU. <http://www.gnu.org/copyleft/copyleft.html>.
- [14] LE PROJET GNU : La page de référence du projet GNU. <http://www.gnu.org>.
- [15] LE PROJET GNU : Le texte de la GNU Free Documentation License. <http://www.gnu.org/copyleft/fdl.html>.
- [16] Le site Internet Musique Libre. <http://www.musique-libre.com/>.
- [17] NETCRAFT : Les statistiques de serveurs webs Netcraft. Internet. http://news.netcraft.com/archives/web_server_survey.html.

- [18] NICOLAS BOUILLON, LUCAS NUSSBAUM et THOMAS PETAZZONI : Le livret du Libre. Internet, <http://www.livretdulibre.org/>, mai 2005.
- [19] Eric S. RAYMOND : *The cathedral and the bazaar : musings on Linux and open source by an accidental revolutionary*. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2001.
- [20] RYXÉO : RyXéo : l'entreprise, les personnes. Internet. <http://ryxeo.com/entreprise.php>.

Annexes

- Code source du Mulot en wxPython
- Documents d'introduction à Python et wxWidgets
- Exercices de wxPython