



Université
Paul Sabatier
TOULOUSE III



TOULOUSE

Informatique



Memobox – 2G Technologies
12 Bv de l'Europe
31850 – MONTRABÉ

Antoine de ROQUEMAUREL

Stage effectué chez Memobox – 2G technologies
Du 10 Avril 2012 au 22 Juin 2012

Pour M. Denis MALLET

Pour M. Patrick MAGNAUD

Rapport de stage

Refonte du système multi-langue pour une application web

Antoine de ROQUEMAUREL

Stage effectué chez Memobox – 2G technologies
Du 10 Avril 2012 au 22 Juin 2012

Pour M. Denis MALLET
Pour M. Patrick MAGNAUD

Rapport de stage

Refonte du système multi-langue pour une application web

Memobox – 2G Technologies
12 Bv de l'Europe
31850 – MONTRABÉ



Remerciements

À l'issue de ce travail de ce stage, je me rend compte que je ne pourrais pas être là ou j'en suis sans certaines personnes, à qui je désire témoigner ma reconnaissance.

Je tiens tout d'abord à remercier l'entreprise Memobox / 2G Technologies de m'avoir accueilli chaleureusement et de m'avoir permis d'y réaliser mon stage et plus particulièrement Denis MALLET, mon maître de stage, pour son accueil et pour me permettre de continuer l'aventure au mois de Juillet.

Ma gratitude va à Romain AURIAC pour m'avoir aidé dans mon projet et m'avoir conseillé tout au long du stage, mais également de m'avoir rapidement intégré dans l'équipe de développement.

Je suis reconnaissant envers Monsieur Patrick MAGNAUD, professeur tuteur, pour me suivre et être venue au cours de ce stage.

Mes remerciements vont également à tous les membres de l'Institut Universitaire de Technologies 'A' de Toulouse, professeurs et membres du service techniques sans qui je ne serais pas là aujourd'hui.

Je remercie Mathieu, qui effectuait son stage dans les locaux de l'entreprise, pour ses conseils, et nos moments de détente toujours sympathiques autour d'un petit café.

Enfin, je remercie Diane pour avoir eu la gentillesse de relire ce rapport.

Table des matières

Remerciements	5
Introduction	9
1 Le groupe Memobox / 2G Technologies	11
1.1 L'historique	11
1.2 La mission de Memobox / 2G Technologies	11
1.3 Les produits de l'entreprise	12
1.4 Organisation dans la société	12
1.5 L'environnement de travail	13
2 Présentation de mon activité	15
2.1 AUDITELcom V7	15
2.2 Mon rôle	16
2.3 Étude de l'existant	16
2.4 Solutions techniques	18
2.5 Réalisation des solutions proposées	19
2.6 Et après ?	27
3 Conclusion	29
3.1 Bilan professionnel	29
3.2 Bilan technique	29
3.3 Bilan personnel	30

Introduction

En fin de deuxième année de DUT¹ Informatique, j'ai dû effectuer un stage de 11 semaines en milieu professionnel afin de compléter ma formation.

Grâce à Madame BENSADOUN, j'ai eu connaissance d'un stage situé dans une entreprise du secteur des télécommunications : Memobox / 2G Technologies, ce stage consistait à la participation au développement d'une application en mode SaaS².

Intéressé par les technologies web, j'ai pris rendez vous avec Denis MALLET et Romain AURIAC afin de savoir en quoi consistait exactement le stage.

Cet entretien a fait grandir mon intérêt pour le poste, et ma candidature a été retenue.

Dans ce rapport de stage, je vais vous présenter mon travail qui a duré 11 semaines, et les connaissances que j'ai pu en retirer, autant sur le point technique que personnel.

Dans une première partie, je vous présenterai donc l'entreprise et le contexte professionnel dans lequel j'ai évolué.

Au début du stage, ma mission s'est précisée : refondre le système de traduction d'une application web multi-langue, ainsi dans une deuxième partie je vous présenterai d'une part l'étude de l'existant, et vous expliquerai pourquoi le système devait être refondue, et d'une autre part mes solutions techniques pour palier à ce problème : un méta-langage et une application externe aidant la traduction.

Enfin, dans la troisième partie, je ferai un bilan personnel et professionnel mettant en valeur ce que ce stage m'a apporté.

1. Diplôme Universitaire de Technologies

2. Software As A Service

Le groupe Memobox / 2G Technologies

1.1 L'historique

1994 : Création de Memobox

À l'origine de Memobox, deux spécialistes, l'un des services de télécommunications : Christophe FORNÈS et l'autre du service réseaux : Denis MALLET.

Ensemble, ils décident en 1994 de développer et commercialiser leur propre ligne d'interfaces de communication pour PBX : EdelBox.

Ces interfaces sont commercialisées auprès des éditeurs de logiciels de taxation et d'analyse de trafic téléphonique.

Memobox est né et ne compte aucun salarié.

1995 : Développement commercial

Poussé par ses premiers clients, Memobox étend sa gamme en créant un système de taxation téléphonique autonome pour les TPE¹ et PME² : Taxabox, décliné également en version hôtelière (Hotelbox). Pour se faire, elle embauche son premier salarié pour étendre ses capacités de développement.

1997 : Le pari du service avec AUDITELcom

À cette date, Memobox compte 3 salariés. Après le succès rencontré auprès des éditeurs de logiciels et intégrateurs en télécoms, Memobox pressant le besoin des entreprises en matière de gestion des coûts télécoms et pour s'adapter à son marché, décide de lancer AUDITELcom le premier service de GFT³ en mode hébergé. Une augmentation de capital accompagne ce nouvel axe de développement.

2003 : Memobox renforce sa position

Le déploiement des 800 premiers sites nationaux du Ministère de l'intérieur est bouclé en 4 mois et le service satisfait pleinement le client. Pour réaliser cette prouesse, Memobox a consolidé ses équipes de développement et d'exploitation. L'effectif est porté à 13 personnes.

2004 : Les deux fondateurs de Memobox rachètent la totalité de l'entreprise

La bonne santé de Memobox en 2004 permet à Christophe FORNÈS et Denis MALLET de racheter à Global Concept la totalité des parts que ce dernier détenait dans Memobox. La société est désormais filiale à 100% du groupe TELECOMS FM⁴ dont Christophe FORNÈS et Denis MALLET sont les deux seuls actionnaires. AUDITELcom compte maintenant plus de 2500 sites raccordés et représente désormais 91% du chiffre d'affaire de la société.

2007 : Memobox renforce sa stratégie

Forte de sa politique de développement, Memobox décide de renforcer sa stratégie :

- En octobre 2007, elle rachète 2G Technologies à sa maison mère, QUESCOM.
- En décembre 2007, elle lève un million d'euros auprès d'ALTO Invest.

1.2 La mission de Memobox / 2G Technologies

Memobox / 2G Technologies est un éditeur de solutions de GFT et de *Call Accounting*. Forts de leur expérience sur le marché, l'entreprise propose à ses clients une gamme complète d'applications, et de services performants et adaptés aux évolutions permanentes du marché des télécoms.

La société possède un large panel de clientèle :

-
1. Très Petites Entreprises
 2. Petites et Moyennes Entreprises
 3. Gestion Financière des Télécoms
 4. *Facilities Management*

- Intégrateurs télécoms,
- Prestataires de services,
- TPE/PME,
- Grandes entreprises,
- Administrations et organismes publics.

La volonté de Memobox / 2G Technologies est également d'être à l'écoute des entreprises pour les conseiller et leur proposer les solutions les mieux adaptées à leur besoins. Dans cette optique, ils proposent de nombreuses alternatives à leurs clients en terme de gammes⁵ mais aussi en terme de modes d'installation⁶.

Ils fournissent aussi des prestations d'administration et d'audit de leurs outils pour permettre aux clients de se concentrer sur leur cœur de leur métier et bénéficier de conseils d'experts avisés.

1.3 Les produits de l'entreprise

1.3.1 Les produits de 2G Technologies

2G Technologies a développé les gammes de logiciels suivants :

GeoTaxe ES Progiciel destiné aux grandes sociétés, comportant une machine hébergée par les clients collectant leurs données pour les mettre à leurs dispositions via leurs infrastructures internet.

Geotel Gestion des appels téléphoniques dans les chambres d'hôtel

Geopital Gestion des appels téléphoniques en milieu médicalisé

Geovox Serveurs vocaux d'entreprise

1.3.2 Les produits de Memobox

Memobox a développé les gammes de logiciel suivants :

AUDITELcom Application sous forme de service destinés aux grands comptes

AUDITELbox Appliance à installer chez le client de petite taille. Il permet la réalisation d'un audit des télécoms en temps réel.

1.4 Organisation dans la société

1.4.1 Mon rôle dans la société

J'ai intégré l'équipe de développement de AUDITELcom, en effet, mon sujet de stage était de refondre le système multi-langues de AUDITELcom. L'équipe de développement du projet était composée de Romain AURIAC et de Denis MALLET, je me suis rapidement intégré à l'équipe et ai apporté ma contribution au projet.

1.4.2 Le personnel du groupe de Toulouse

Denis Mallet Vice-président et Directeur Technique

Jean-Marie Lagarde Responsable R&D

Philippe Dureux Ingénieur télécoms

Romain Auriac Ingénieur développement Web

5. logiciels pour PME, grands comptes, hôtels, hôpitaux

6. appliances, logiciels ou services externalisés

Mathieu Soum Stagiaire dans le développement logiciel

Antoine de Roquemaurel Stagiaire dans le développement web

L'organigramme est disponible figure 1.1.

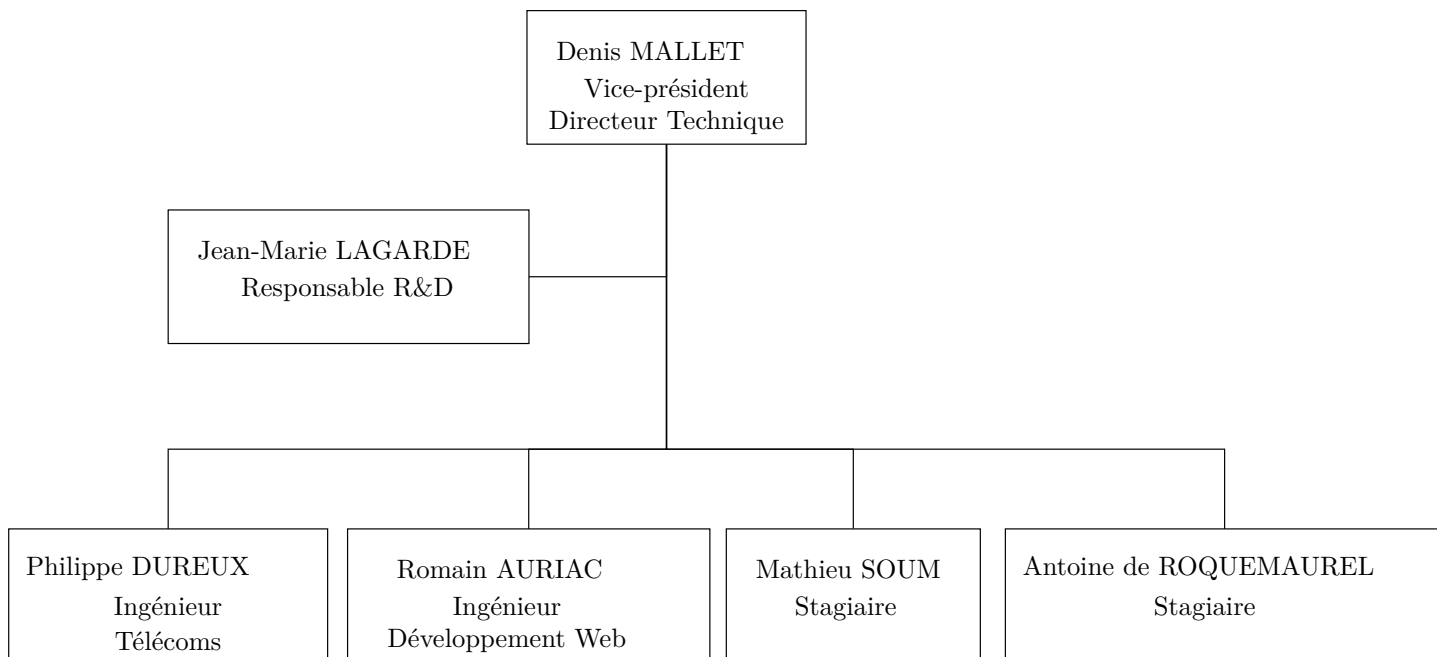


FIGURE 1.1. Organigramme du service R&D à Toulouse

1.4.3 Les horaires de travail

Les horaires de l'entreprise sont assez souples, cependant afin d'assurer un standard téléphonique, tous les employés doivent être présent entre 9h30 et 17h30.

Pour ma part, j'arrivais le matin vers 9h30, allais déjeuner vers 12h30 et reprenais le travail de 14h jusqu'à 18h30.

1.5 L'environnement de travail

1.5.1 Mon poste de travail

Je travaillais sous un environnement Windows XP Professionnel SP3, ce système était installé sur un ordinateur de série Dell, munit d'un processeur Intel Core 2 Duo cadencé à 2Ghz avec 2Gio de RAM.

Ce poste possédait deux écrans, ainsi je pouvais travailler plus vite en ayant des applications répartis sur les deux écrans.

1.5.2 Le serveur web

Afin de pouvoir effectuer des tests, j'utilisais un serveur distant, situé dans les locaux de l'entreprise. Ce serveur, nommé `to-audidev-mbx` était un serveur FreeBSD, avec un serveur Apache et une base de données MySQL. Ce

serveur n'était destiné qu'au développement, ainsi nous avions une base de données permettant les tests éventuels. Une fois que des modifications stables ont été effectuées et commitées, celles-ci étaient basculées en production, sur un serveur situé à Paris.

1.5.3 Pour le développement

J'ai utilisé l'environnement de Développement Intégré⁷ Eclipse, il a l'avantage d'être gratuit et très modulable ; grâce aux nombreux projets qui ont été réalisés pour cette plateforme. Dans l'entreprise tout le monde l'utilisait, que ce soit pour du PHP ou du C++. En ce qui me concerne, j'avais un environnement PHP, je possédais le projet *PHP Development Tools*⁸. Il permet de faire de l'analyse syntaxique, de l'autocomplétion ainsi que la navigation entre les classes et méthodes.

1.5.4 Pour le travail collaboratif

N'étant pas seul sur le projet, il a fallu mettre en place des outils permettant le travail collaboratif. Pour cela tous les projets de la société sont regroupés sous un système de gestion de versions, en l'occurrence, SubVersion. Ce système permet d'enregistrer toutes les modifications effectuées sur un fichier. Ainsi chaque collaborateur peut mettre à jour, comparer ou récupérer une ancienne version d'un fichier. Pour avoir un environnement Eclipse complet, il y a été rajouté le projet *subclipse* proposé par Tigris.org. Il contient *svnClientAdapter* et *SVNKit*.

1.5.5 Pour la synchronisation serveur

Le site de développement étant sur un serveur distant, il fallait pouvoir synchroniser mes modifications avec ce serveur, ainsi tous les développeurs web utilisaient un plugin eclipse appelé FileSync qui synchronisait nos fichiers avec le serveur à chaque enregistrement du fichier.

7. EDI

8. PDT

Présentation de mon activité

2.1 AUDITELcom V7

Durant ce stage, j'ai travaillé sur le produit AUDITELcom qui est actuellement en version V7.

AUDITELcom est une plateforme complète de TEM¹ particulièrement adaptée au contexte des Grands Comptes. Elle équipe également des fournisseurs de services soucieux d'apporter à leurs clients des solutions éprouvées à forte valeur ajoutée. Entièrement modulaire, elle s'adapte aux souhaits et aux contraintes des clients et permet de gérer l'ensemble du budget et des ressources télécoms : voix, data fixes et mobiles.

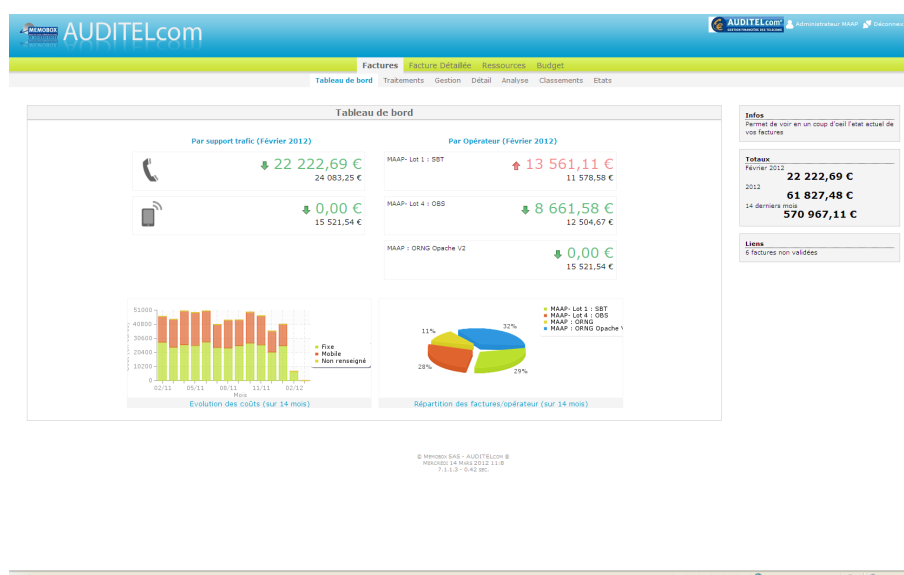


FIGURE 2.1. AUDITELcom V7

AUDITELcom est un service en ligne qui, de part son architecture modulaire et sa forte industrialisation, s'adresse aussi bien aux PME² qu'aux Grands Comptes et Très Grands Comptes.

Fonctionnant en mode SaaS³, Cette plateforme peut être installée chez un client ou mutualisée dans les locaux d'un fournisseur de services ou dans ceux de Memobox / 2G Technologies.

Elle assure une qualité irréprochable des données, chose essentielle pour une solution de TEM :

- Les grilles tarifaires des opérateurs sont en permanence à jour,
- Les nouveaux PABX et IPBX sont toujours pris en compte,
- Les nouvelles fonctions statistiques sont mise en ligne pour tous,
- Les nouvelles fonctionnalités sont automatiquement activées.

L'offre de service AUDITELcom délivrée sous forme d'un abonnement, est une solution clé en main permettant de ne pas investir en matériel et logiciel, et de ne pas monopoliser de ressources et de compétences particulières.

1. Telecom Expense Management

2. Petites et Moyennes Entreprises

3. Software as a Service

2.2 Mon rôle

Lors de mon stage, j'ai participé au développement de AUDITELcom. J'ai intégré l'équipe de développement composée de Romain AURIAC et Denis MALLET, le système étant en production, il fallait faire attention à n'effectuer aucune régression de l'application, et effectuer suffisamment de tests avant que mes modifications ne soient apportées sur le serveur de production afin que les clients ne subissent aucune préjudice.

Le développement d'AUDITELcom fut commencé dans les années 2000 et plusieurs développeurs ont apporté leur contribution, ainsi, on pouvait distinguer différents styles de programmation, certaines portions de code pouvant être vieillissantes.

Lors de mon arrivée, AUDITELcom était destiné à être multilingue, ainsi il était possible de se connecter en Français, Anglais ou Espagnol. Cependant le système qui permettait à l'application d'être multilingue n'était pas parfait, et rendait difficile la maintenance et la traduction de l'application. En effet, lorsque nous nous connectons pour visualiser l'application en Anglais, la traduction n'était pas effective partout. Ainsi, certaines phrases s'affichaient en français. Cependant la traduction n'allait pas continuer tant que le système n'était pas viable.

Ainsi, mon travail fut d'étudier le système déjà développé, afin d'en analyser les erreurs qui ont été faites pour pouvoir concevoir un nouveau système qui serait beaucoup plus stable et pérenne et rendrait le travail de traduction le plus simple et rapide possible.

2.3 Étude de l'existant

2.3.1 Le système

Pour son système de traduction l'application possédait une classe **TTranslator**, toutes les traductions devaient passer via cette classe à l'aide d'un singleton, la classe **TTranslator** faisait ensuite des appels à la classe **TDataLinkAbstract** pour faire le lien avec la base de données contenant toutes les traductions.

2.3.1.1 La table RES_Translations dite Ancienne Table

Toutes les traductions étaient stockées dans la table **RES_Translations**, ci-dessous sa structure.

RES_Translations
<u>ID_RES</u> : String
<u>Language</u> : Char
TranslatedText : String
Valide : Boolean

FIGURE 2.2. Table RES_Translations

ID_RES Le nom de la constante de traduction.

Language La langue de traduction en ISO-639 (fr, en ou es)

TranslatedText La valeur de la constante de traduction pour la langue donnée

Valide Valide ou non, 1 si la traduction est correcte, 0 sinon.

2.3.1.2 La méthode `getTranslation`

La classe `TTranslator` possédait une méthode `getTranslation(string, string)`, le premier paramètre était une constante de traduction et le deuxième paramètre, optionnel, correspondait à la langue de traduction. La constante de traduction permettait de faire le lien avec la base de données, chaque entrée de la base de données était caractérisée par un ID et une Langue.

Ainsi, lorsqu'un développeur souhaitait traduire un texte, il devait connaître la constante de traduction, si celle-ci n'existait pas, il devait ajouter une entrée dans la base de données.

```
<?php
$translator = TTranslator::singleton();
echo $translator->getTranslation('LANG_DEFAULT');
?>
```

Listing 2.1. Exemple d'appels de `getTranslation`

La langue était optionnel étant donné que par défaut, `TTranslator` utilisait la langue du navigateur, ou la langue demandée lors de la connexion.

2.3.2 Avantages et inconvénients du système

Afin de pouvoir développer un système le plus robuste possible, j'ai analysé l'existant afin d'en tirer les avantages et inconvénients. eux-ci sont disponibles table 2.1.

Avantages	Inconvénients
<ul style="list-style-type: none"> – L'utilisation d'une base de données permet une organisation simple et relationnelle des informations – Convention de nommage. Le préfixe <code>LANG_</code> pour les ID permet de retrouver facilement une constante dans le code en cas de recherche 	<ul style="list-style-type: none"> – Duplication de constante, cela vient sûrement des développeurs qui ne trouvant pas une constante, en créaient une autre. – Traduction de certaines variables, comme par exemple 13 derniers mois – Si la constante n'était pas traduite dans la langue demandée, le visiteur verrait un ID, ce qui n'est pas très compréhensible – Recherche, insertion et traduction difficiles.

TABLE 2.1. Avantages et inconvénients du système

2.3.3 Le problème

Lorsque je suis arrivé, le système de traduction devenait difficile à utiliser. La base de données contenait beaucoup de redondances, par exemple la base contenait 7 ID différents ayant chacun pour valeur *Coût*.

Également elle contenait 378 champs qui n'étaient pas valide, ce nombre assez important rendait le travail de validation beaucoup trop conséquent. Il devenait urgent de trouver un nouveau système.

Ce problème est arrivé en partie à cause des développeurs qui utilisaient le système, il ne pouvait pas facilement savoir si une constante était déjà présente pour ce qu'il voulait faire, ainsi il leurs était plus simple et plus rapide d'en créer une nouvelle, rapidement la base de données devint incontrôlable. Certains ont également choisis de ne pas créer d'ID, et de mettre leur texte directement dans l'appel de la méthode, cette fois ci ce fut le code du projet qui devint difficile à maîtriser.

2.4 Solutions techniques

Une fois le système étudié, j'ai réfléchi à des solutions qui pourraient permettre d'avoir un système perenne, simple à utiliser et qui éviterait les redondances d'informations.

2.4.1 Un méta-langage

Pour éviter la redondance d'informations, j'ai choisi de créer un méta-langage qui serait utilisé dans la base de données, celui-ci permettrait plusieurs choses :

La concaténation de constante Pour éviter de traduire toujours les même mots qui reviennent souvent, il paraissait intéressant de pouvoir intégrer un ID de constante dans un texte, ainsi les traducteurs n'auront à traduire qu'une seule fois la dite constante

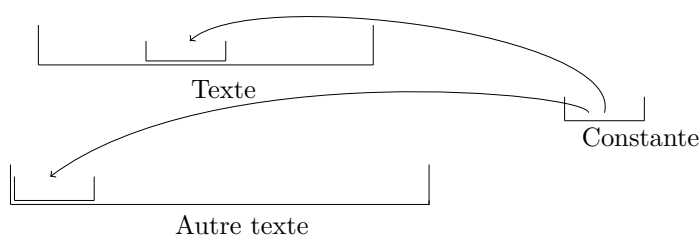


FIGURE 2.3. Schéma des constantes

L'ajout de variables Il y a souvent des phrases qui reviennent avec seulement une partie du texte qui change, cette partie pourrait être un nombre comme dans 13 derniers mois, mais cela pourrait également être une chaîne de caractères comme dans "appel fixe" ou "appel mobile", en effet, seul le dernier mot change.

Une gestion du pluriel singulier L'utilisation de variables numériques va ainsi soulever un problème qui est le singulier ou le pluriel, en effet, comment faisons nous si nous avons 13 lignes fixes ou 1 ligne fixe ? Ce méta-langage permettra de choisir comment s'effectue cet accord en fonction de la variable située devant.

Des exemple de ce méta-langage sont disponible section 2.5.1.2 page 19.

2.4.2 Une application externe ou client lourd

La solution précédente semble indispensable afin de prévenir la duplication d'informations, cependant, le problème des développeurs persistera toujours, ils trouveront difficile la recherche, et préféreront créer une nouvelle constante, et nous tomberons dans les même travers que précédemment.

Ainsi, j'ai choisi de développer une application qui permettrait de palier à ses problèmes. Celle-ci doit être simple d'utilisation, le développeur doit effectuer rapidement ce qu'il veut et le logiciel doit lui faire gagner du temps afin qu'il soit utilisé.

Cette application permettrait donc les choses suivantes :

Rechercher avec un mot clé ou un ID La fonctionnalité principale permettra au développeur de chercher simplement si une constante existe, si celle-ci existe, il pourra copier dans le presse papier le presse-papier d'appel de la méthode afin de lui simplifier le travail.

Ajouter une constante Si la constante recherchée n'existe pas, il devra être possible d'ajouter une nouvelle constante.

Éditer une constante Il peut arriver qu'une erreur se glisse dans une constante, ainsi il sera possible de la modifier.

Traduire une constante existante Quand une constante n'est pas traduite, le développeur, ou un traducteur pourra ajouter la traduction des langues manquantes.

Basculer une constante de l'ancienne table vers la nouvelle Étant donné tous les problèmes de la table `RES_Translations`, nous avons choisi de créer une nouvelle table `RES_dicoLanguage`, cependant, des informations de l'ancienne table sont tout de même valables, ainsi il sera possible de demander à basculer un ID de l'ancienne table sur la nouvelle

2.5 Réalisation des solutions proposées

2.5.1 La nouvelle classe TLanguage

Pour pouvoir effectuer la solution proposée section 2.4.1, j'ai créé une nouvelle classe PHP, appelée `TLanguage`. J'ai choisi de garder les parties fonctionnelles de l'ancien système, c'est-à-dire la détection et le choix de la langue du visiteur, j'ai cependant redéveloppé toute la méthode `getTranslation`, afin que celle-ci soit capable d'interpréter le meta-langage

2.5.1.1 La base de données, la table `RES_dicoLanguage` dite nouvelle table

La base de données existante était bien conçue, cependant, mon meta-langage demandait à ajouter un nouveau champ : `Type`. En effet, nous pouvons inclure dans une phrase un ID, afin de ne pas traduire plusieurs fois la même chose. Il faut faire la différence entre les items, qui sont des mots qui pourront être incluse dans d'autres phrases, et les phrases qui ne doivent jamais être incluse, ceci afin d'éviter une éventuelle récursivité pour garder une certaine logique.

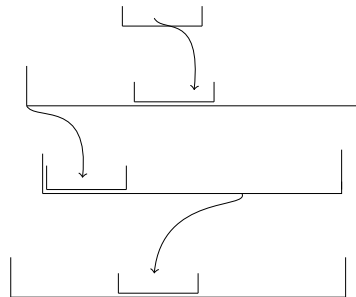


FIGURE 2.4. Le problème qui sera évité avec les items et les phrases.

Ainsi, la nouvelle table devient donc :

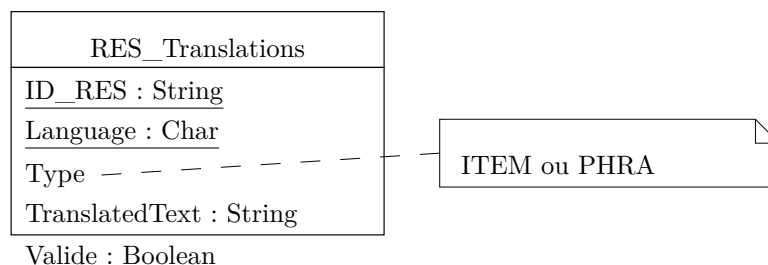


FIGURE 2.5. Table `RES_dicoLanguage`

2.5.1.2 La méthode `getTranslation`

J'ai redéveloppé intégralement cette méthode afin de pouvoir y intégrer un parseur qui interprète le méta-langage. Ainsi, une fois développé il est possible, comme expliqué section 2.4.1 de :

Concaténer des constantes Pour cela, il suffit au développeur d'ajouter dans la valeur de traduction [NOM_ID] avec NOM_ID ayant pour valeur l'ID d'une constante de traduction. Les crochets seront ainsi remplacés par la constante en question.

Si la constante n'a pas été traduite pour la langue du visiteur, celle-ci sera affichée en Français.

ID_RES	TranslatedText	Type_RES
TRANS_GENERAL_DATA	Données [TRANS_GENERAL]	PHRA
TRANS_GENERALES	Générales	ITEM

TABLE 2.2. Exemple de constante – Base de données

```
<?php
$language = TLanguage::singleton();
echo $language->getTranslation('TRANS_GENERAL_DATA');
?>
```

Listing 2.2. Exemple de constante – Code

Données Générales

EXEMPLE 2.1 – Utilisation des constantes

Ajouter des variables Il est également possible d'ajouter des variables dans une valeur, ainsi, dans le champ de la base de données, le développeur doit mettre <variable>, *variable* pouvant être remplacée par ce qu'il veut. Cela permet de donner un nom à sa variable pour que les personnes trouvant la constante comprennent à quoi correspond celle-ci. Lors de l'appel de la méthode `getTranslation` le développeur fera passer la liste des arguments nécessaires au bon fonctionnement de la traduction.

ID_RES	TranslatedText	Type_RES
TRANS_PBX_NO_LINES	<nbPBX> PBX sans ligne ni poste	PHRA

TABLE 2.3. Exemple de variables – Base de données

```
<?php
$language = TLanguage::singleton();
echo $language->getTranslation('TRANS_PBX_NO_LINES', 4);
?>
```

Listing 2.3. Exemple de variables – Code

4 PBX sans ligne ni poste

EXEMPLE 2.2 – Utilisation des variables

Gérer le pluriel ou le singulier Si le développeur a mis des variables qui attendent un nombre, il doit signaler les accords éventuels qui suivent la variable avec la syntaxe (**sing|plur**).

ID_RES	TranslatedText	Type_RES
TRANS_NB_RINGINGS	<nbSonneries> sonneri(e es)	PHRA

TABLE 2.4. Exemple avec du pluriel/singulier – Base de données

```
<?php
$language = TLanguage::singleton();
echo $language->getTranslation('TRANS_NB_RINGINGS', 15);
?>
```

Listing 2.4. Exemple avec du pluriel/singulier – Code

15 sonneries

EXEMPLE 2.3 – Utilisation du pluriel et du singulier

2.5.2 Le client lourd : MemoLanguage

Une fois le nouveau moteur de traduction développé, il me fallait concevoir un outil de traduction aidant le développeur comme expliqué section 2.4.2 page 18.

Cet outil fut baptisé **MemoLanguage**, à mon départ, il en était à sa version 0.9.

2.5.2.1 Méthode Agile

Afin d'avoir un client lourd qui soit le plus adapté aux besoins possibles, nous avons choisi une méthode Agile, par itération successive. Ainsi, une fois qu'une première version de **MemoLanguage** était finie, j'ai mis le logiciel en production, l'utilisation du logiciel a permis de rapidement ajouter de nouvelles fonctionnalités afin de satisfaire au maximum les besoins de l'entreprise.

Le logiciel a été développé en 5 itérations successives, les résultats qui suivent sont les résultats obtenus après 4 itérations. La 5ème est disponible section 2.5.3 afin que vous puissiez voir un exemple d'itération.

2.5.2.2 Technologies choisies

Afin d'avoir l'outil le plus pratique possible, celui-ci devait être installé en natif chez chacun des développeurs, en effet il était plus pratique pour les développeurs d'avoir une application dans la barre des tâches. Une page web n'aurait pas été pratique, en effet, le développeur aurait dû changer d'onglet régulièrement, alors qu'une application native peut être mise dans un coin, ou réduite au besoin du développeur.

Ainsi, la technologie choisie fut **Adobe Air**⁴, c'est une machine virtuelle multi-plateforme qui s'exécute sur le système d'exploitation.



Il est ainsi possible d'utiliser des technologies web pour créer un client natif, ce qui permet d'avoir la puissance d'une application native avec la souplesse des technologies web.

4. Adobe Integrated Runtime



Pour effectuer des recherches sur un serveur distant, avoir une bonne ergonomie et ne pas avoir à recharger la page continuellement, j'ai choisi d'utiliser du JavaScript pour développer le client, sa puissance me permettant d'interroger le serveur.

Je l'ai couplé avec du XHTML⁵ et du CSS⁶ pour l'affichage et la mise en forme.

C'est donc avec la combinaison de ces technologies que j'ai utilisé l'AJAX⁷

Il fallait cependant également développer un serveur qui ferait le lien entre le client Adobe Air et la base de donnée, cette base de données étant celle d'AUDITELcom, c'est une base de données MySQL, ce qui permet d'avoir une base facilement interfaçable avec Apache et PHP tout en gardant de bonnes performances.



Le serveur sera donc codé en PHP Orienté Objet. Cela permettra d'avoir le maximum de puissance, mais également un code propre, organisé et réutilisable.

Ainsi, nous avons une architecture client serveur, cette architecture est disponible figure 2.6.

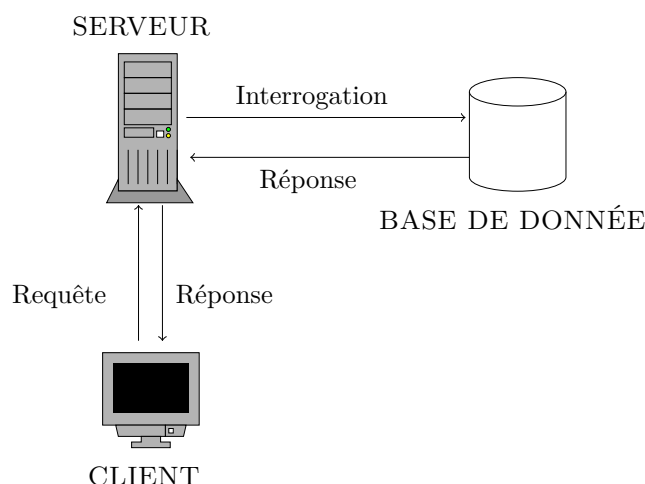


FIGURE 2.6. Architecture client – serveur

2.5.2.3 Développement du serveur

Le serveur est destiné à être appelé via le JavaScript, donc juste avec l'URL, ainsi toutes les variables seront dans l'URL.

J'ai donc choisi une architecture où toutes les requêtes s'articulent autour d'un fichier index.php, avec plusieurs variables en URL.

5. eXtensible HyperText Markup Language

6. Cascading Style Sheets

7. Asynchronous Javascript and XML

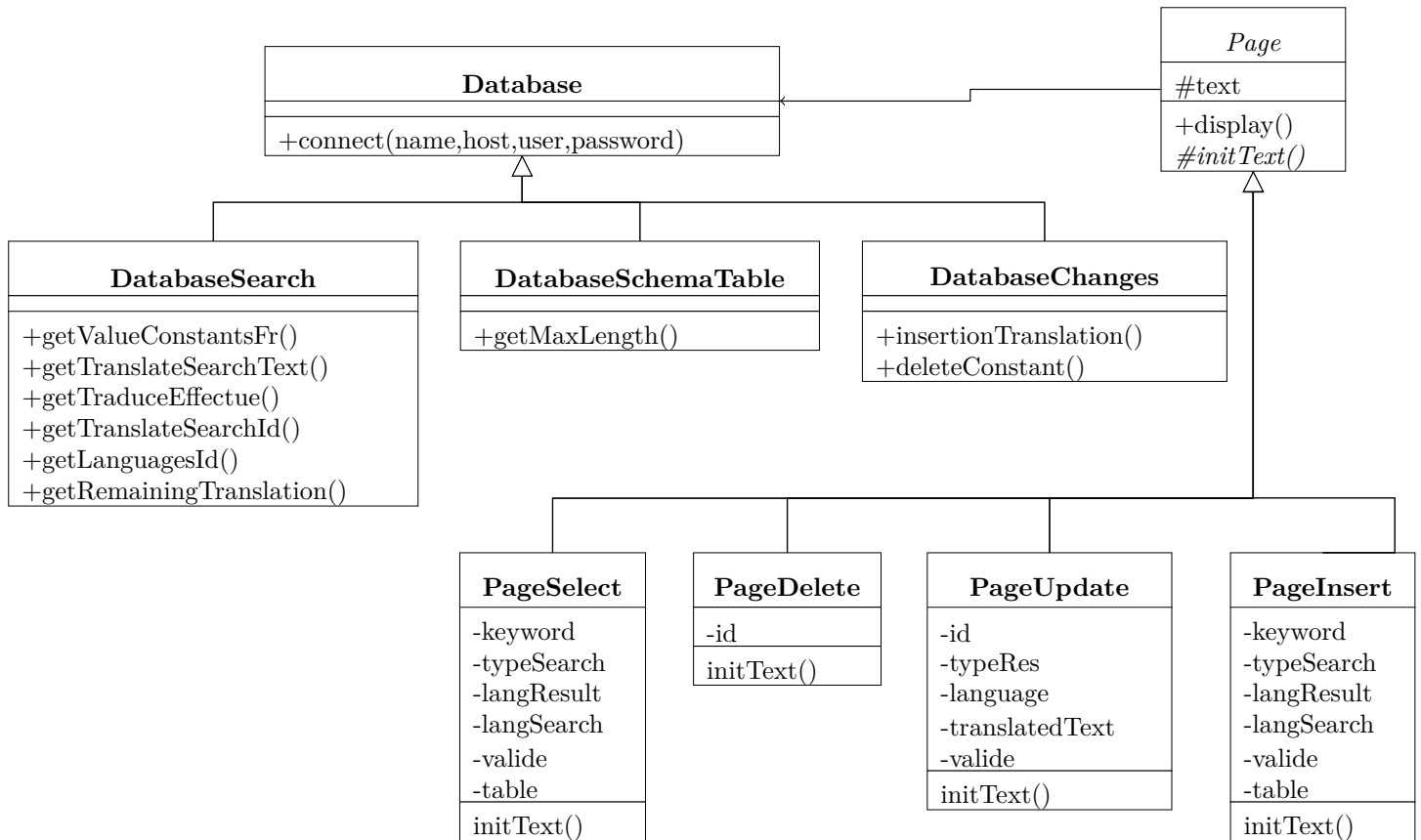


FIGURE 2.7. Diagramme de classe du serveur

Sur la figure 2.7 est disponible le diagramme de classe du serveur, dans ce diagramme de classe, les méthodes privées ne sont pas représentées pour simplifier le diagramme.

Recherche Pour effectuer une recherche, il faut appeler l'URL `index.php?page=select`, cela aura pour effet de construire une classe `PageSelect`.

Les autres informations sont transmises via l'URL avec les variables suivantes

type – Le type de recherche. Celui-ci peut-être par ID ou par **text**

k – Le ou les mots clefs. S'il y a plusieurs mots clefs ils doivent être séparés par des tirets (-).

langSearch – La langue de recherche, cela n'est utile qu'en cas de recherche par texte.

langResult – La ou les langues d'affichage, si plusieurs langues sont renseignées celles ci doivent être séparées par des pipes (|).

valide – Affichage des champs valides. Valide=1 pour afficher les champs Valide, Valide=0 pour afficher les non valides et Valide=2 pour tout afficher.

table – Le nom de la table dans laquelle on cherche, cela permet de pouvoir chercher dans la nouvelle table ou dans l'ancienne suivant notre convenance.

La classe `PageSelect` effectuera ensuite une recherche à l'aide de la classe `DataBaseSearch`, qui est en charge d'effectuer les différentes requêtes pour la recherche.

J'ai ainsi développé mon protocole de communication avec le client. Si des informations sont trouvées, elles seront affichées à l'écran, chaque colonne de la base est séparée par trois tirets (---), chaque ligne est séparé par trois plus (+++).

Également, entre trois dièses (###) sont affichés les langues de traduction qui existent pour cette constante. Chaque langue est séparés par des espaces () au sein des dièses.

Si aucune donnée n'a été trouvée dans la base pour cette recherche, l'API⁸ retournera NOT_FOUND.

Le client pourra ensuite utiliser les séparateurs pour organiser les informations.

```
index.php?p=select&type=text&k=Ajouter&langSearch=fr&langResult=fr|en&
valide=1&table=RES_dicoLanguage
```

Affichage :

```
TRANS_NAME --- en###en es fr ### --- 1 --- ITEM --- Name+++TRANS_NAME ---
fr###en es fr ### --- 1 --- ITEM --- Nom+++TRANS_NB_CONTENT_USERS ---
fr###fr ### --- 1 --- PHRA --- Nombre de contenant des utilisateurs
+++TRANS_NB_POSTES --- fr###fr ### --- 1 --- PHRA --- Nombre de poste
+++TRANS_NB_USERS --- fr###fr ### --- 1 --- PHRA --- Nombre d'utilisateurs+++
```

EXEMPLE 2.4 – Exemple de recherche

Modifications La modification s'effectue avec la page `index.php?page=update` quiinstanciera `PageUpdate`.

Pour effectuer la modification, il faut passer en paramètre l'ID à modifier et toutes les nouvelles valeurs de la table.

ID – L'ID à modifier dans la table

type – Le nouveau type (ITEM si c'est un item ou PHRA si c'est une phrase)

language – La nouvelle langue de la valeur

value – La valeur correspondante

valide – Si le champ est valide ou pas (1 valide, 0 sinon)

Tous ces paramètres doivent être donnés, si un champ ne doit pas être modifié, il faut tout de même signaler la valeur correspondante.

Si l'édition s'est bien passée, le serveur affichera GOOD_UPDATE.

```
index.php?p=update&id=TRANS_NAME&type=PHRA&language=fr&value=Nom&valide=1
```

EXEMPLE 2.5 – Exemple d'édition

Insertion L'insertion s'effectue exactement de la même manière que la modification à la différence que l'URL appelée est `index.php?page=insertion` afin d'appeler le constructeur de `PageInsert`

Si l'insertion s'est bien déroulée, le serveur renverra GOOD_INSERT.

```
index.php?p=insert&id=NEW_CST&type=PHRA&language=fr&value=un%20test&valide=1
```

NOTA : le symbole %20 sera remplacé par un espace ().

EXEMPLE 2.6 – Exemple d'insertion

8. Application programming interface

Suppression La suppression s’effectue avec la page `index.php?page=delete` quiinstanciera `PageDelete`.

La suppression est relativement simple, il suffit de donner en paramètre l’ID à supprimer, et toutes les traductions associées seront perdues définitivement. Il faut donc faire attention quand on utilise la suppression.

ID – L’ID à supprimer définitivement de la table

Si la suppression s’est bien passé, le serveur affichera `GOOD_DELETE`.

```
index.php?p=delete&id=TRANS_NAME
```

EXEMPLE 2.7 – Exemple de suppression

2.5.2.4 Développement du client lourd : MemoLanguage

Une fois le serveur développé, il fallait maintenant faire le client, celui-ci est codé en JavaScript. Il permet d’effectuer une recherche, une édition, une suppression, une traduction, un ajout de constante. Beaucoup de fonctionnalités ont été ajoutées afin de faciliter le travail des développeurs et pour les encourager à utiliser l’outil.

Toutes ces fonctionnalités sont possibles grâce au serveur, j’effectue donc des requêtes en Javascript avec `XMLHttpRequest`, ce qui me permet de contrôler le serveur et d’obtenir le résultat d’une page dans une chaîne de caractère.

Recherche Afin d’avoir des recherches les plus poussées possible, plusieurs fonctionnalités ont été ajoutées :

Chercher par ID ou Texte – Le développeur peut choisir de chercher directement par ID ou par mots clefs..

Chercher dans l’ancienne ou la nouvelle table – Il est possible de chercher soit dans l’ancienne table, soit dans la nouvelle, afin de pouvoir faciliter la migration des données. Si une recherche est infructueuse dans la nouvelle table, automatiquement l’outil cherchera dans l’ancienne s’il n’y a pas une correspondance. Si c’est le cas, le développeur pourra faire basculer une constante de traduction et toutes ses traductions associées dans la nouvelle table.

Mots entiers – Il peut arriver que l’on ne cherche que des mots entiers, par exemple en cherchant ”nom“ on ne souhaiterait pas obtenir des occurrences pour ”nombre“.

Expression régulière – Si un développeur veut utiliser des recherches complexes, cela est possible à l’aide d’expressions régulières.

Traduction valides – Il est également possible de ne chercher que dans les traductions valides, non valides ou les deux.

Chercher dans une autre langue – Le développeur pourrait souhaiter effectuer une recherche en anglais comme ”name“ ou en espagnol avec ”nombre“. Cependant, si la traduction à une constante n’existe pas pour la langue de recherche, celle-ci ne pourra jamais s’afficher.

Afficher les langues souhaitées – Le développeur peut choisir quelle langue il souhaite afficher, il peut combiner plusieurs langues. Les langues disponibles sont Anglais, Français et Espagnol.

Une capture d’écran d’une recherche est disponible dans le dossier annexe [A.1.1](#) page 7, sur cette capture, on peut voir les différentes cases à cocher permettant d’affiner la recherche.

Également, 4 onglets sont présents, suivant l’onglet sur lequel on se trouve, la recherche s’effectuera dans la nouvelle table ou dans l’ancienne table, comme indiqué par l’onglet.

La présence de plusieurs onglets, permet de ne pas effacer une recherche qui est intéressante.

Édition Les développeurs peuvent éditer des constantes, pour cela lors d'une recherche, ils doivent cliquer sur le bouton éditer de la ligne correspondante.

Une capture d'écran est disponible annexe [A.2](#) page 8.

Suppression Pour supprimer une constante et toutes les traductions associées, il faut cliquer sur le bouton supprimer situés dans la colonne de droite. Une boîte de confirmation s'affiche à l'écran pour éviter les suppressions par inattention.

Une capture d'écran est disponible annexe [A.1.4](#) page 9.

Traduction Un utilisateur pourra traduire une constante qui n'a pas été traduite dans la langue voulue, pour cela, si en cherchant il trouve une constante qui n'est pas traduite dans les trois langues (Anglais, Français et Espagnol), il pourra, s'il le souhaite ajouter une traduction manquante.

Une capture d'écran est disponible annexe [A.1.6](#) page 10.

Ajout de constante Pour ajouter une constante l'utilisateur doit faire fichier puis nouveau (ou contrôle + N), il aura alors un formulaire à remplir pour ajouter sa constante.

Il sera également invité à ajouter une constante en cas de recherche infructueuse.

Une capture d'écran d'un ajout de constante est disponible annexe [A.1.3](#) page 8.

Basculer une constante de l'ancienne base vers la nouvelle Afin d'intégrer au mieux le nouveau moteur de traduction, et d'avoir une base solide le plus rapidement possible, beaucoup de constantes de l'ancienne table doivent être migrés dans la nouvelle, mais cela doit se faire manuellement afin de pouvoir maîtriser la nouvelle table.

Pour cela, **MemoLanguage** permet de basculer une constante de traduction et les traductions associées qui appartenaient à l'ancienne table, vers la nouvelle table.

Une capture d'écran est disponible dans le dossier annexe [A.5](#) page 9, sur cette capture, on remarque que automatiquement un nouvel ID est proposé afin de garder la cohérence avec les préfixes, cependant une invitation à vérifier cet ID est affichée.

2.5.3 5ème itération de MemoLanguage

Un exemple de nouvelle itération fut d'ajouter deux nouveaux champs dans la table **RES_dicoLanguage**.

En effet, pour certaines constantes, il était indispensable d'avoir une abréviation de la valeur de traduction, afin de pouvoir par exemple la mettre en titre de colonne, la valeur de traduction classique pouvant être affichée en infobulle.

A l'inverse, certaines traductions pouvaient avoir le besoin d'une valeur beaucoup plus longue et détaillée.

Figure 2.8 est disponible la nouvelle table `RES_dicoLanguage`.

RES_dicoLanguage
<u>ID_RES</u> : String
<u>Language</u> : Char
Type : Char
TranslatedText_short : String
TranslatedText : String
TranslatedText_long : String
Valide : Boolean

FIGURE 2.8. Table `RES_dicoLanguage`

Cette modification de la base de donnée, a ainsi induit la modification d'une part de la classe `TLanguage`, afin de pouvoir préciser quelle valeur de constante doit être affichée, via la méthode `getTranslation` : pour cela, si la valeur courte ou la valeur longue doit être utilisée, il faut préfixer l'ID de la constante avec `#SHORT#` ou `#LONG#`. Si rien n'est précisé, ou si le champ demandé est vide, ce sera le champ classique qui sera affiché.

```
<?php
$language = TLanguage::singleton();
echo $language->getTranslation('#SHORT#TRANS_NUM');
?>
```

Listing 2.5. Exemple de variables courtes ou longues

EXEMPLE 2.8 – Appel de l'ID courte de `TRANS_NUM`

D'autre part, j'ai dû apporté des modifications à `MemoLanguage`, afin que les développeurs puissent insérer des constantes longues ou courtes, mais également faire des recherches soit dans les constantes longues, normales ou courtes.

Ainsi, le serveur et le client ont été modifiés, le serveur insère dorénavant des valeurs dans les deux nouveaux champs : `TranslatedText_short` et `TranslatedText_long`.

Le client s'interface avec le serveur pour l'insertion et l'édition, et permet l'affichage des différents types de constantes. Il est également possible de choisir dans quel champ s'effectue la recherche.

De nouvelles captures d'écrans de `MemoLanguage` sont disponibles dans le dossier annexes [A.1.7](#) page 10.

2.6 Et après ?

Mon stage est terminé, cependant, mon travail va continuer d'exister.

En effet, `MemoLanguage` sera utilisé régulièrement par les développeurs de AUDITELcom afin d'ajouter des constantes dans le code, les développeurs ont déjà commencé à s'en servir.

Également, des améliorations pourront être apporté au logiciel par d'autre développeur, par exemple une amélioration pourrait être de faire passer mon protocole de communications vers JSON afin de respecter les standards du web. Tout autre amélioration qu'ils trouveront utile en utilisant le logiciel pourra également être apporté afin qu'il

soit le plus pratique possible, c'est pour cela que j'ai développé **MemoLanguage** afin qu'il soit le plus compréhensible possible et qu'il puisse être amélioré sans trop de problème, j'ai donc bien commenté mon code.

Également, je n'ai pas fini d'intégrer le nouveau moteur de AUDITELcom, cependant, je suis embauché au mois de Juillet dans l'entreprise, ainsi je finirais ce travail, afin qu'un seul moteur de traduction ne soit utilisé.

Conclusion

À l'issue de ces 11 semaines de stage passées dans l'entreprise Memobox / 2G Technologies, je souhaite faire un bilan professionnel, technique mais également personnel de ce que m'a apporté mon travail au sein de l'équipe de développement de AUDITELcom.

Dans les pages qui suivront cette conclusion, vous pourrez observer mon Curriculum Vitae avant et après le stage afin de mieux observer ma progression. Un bilan de compétences et également disponible.

3.1 Bilan professionnel

Ce stage fut ma première expérience professionnelle, cela m'a permis de voir que cela n'avait rien à voir avec l'école.

En effet, j'ai découvert qu'en entreprise, nous n'avons pas le droit à l'erreur, je travaillais sur AUDITELcom qui est actuellement utilisé par des clients, ainsi, mes modifications passaient en production, il fallait donc que je teste bien afin que les clients n'aient pas de problème lors de l'utilisation de l'application.

Également, j'ai appris qu'en entreprise, il faut vraiment prévoir tous les cas d'utilisations, qu'un seul cas oublié peut être catastrophique pour le produit.

Enfin, je me suis rendu compte que dans le monde du travail, les programmes sont loin d'être récents, ainsi le développement de AUDITELcom ayant été commencé dans les années 2000, j'ai eu du mal à comprendre certaines parties de code, j'ai également découvert du code mort, ce qui ne facilite pas la compréhension du programme. J'ai donc essayé de développer mon application de façon à ce qu'elle soit simple et facile à comprendre, cependant je sais que dans 10 ans, mon code sera vieux, difficile à comprendre et à maintenir.

Même après mon départ, mon travail sera utilisé et amélioré, cela me fait plaisir, et change de l'école où une fois finis on entend plus parler de notre travail, cependant c'est également plus difficile étant donné qu'il faut développer notre application de façon à ce qu'elle puisse être continuée facilement.

3.2 Bilan technique

D'un point de vue technique, ce stage m'a permis d'apprendre de nouvelles technologies que je ne connaissais pas, Adobe AIR d'une part, je ne connaissais pas son existence, et cela m'a montré qu'avec des technologies web, il est de nos jours possible de faire tout ce que l'on veut.

J'ai aussi appris le JavaScript qui ne nous est pas enseigné à l'IUT, mais également le PHP orienté objet, cependant ma formation en Algorithmique et en technologies web m'a permis de rapidement apprendre ces nouvelles technologies.

Enfin, ce stage m'a également montré que l'optimisation dans le cadre du web est très importante, en effet nous ne savons pas d'avance combien de personnes se connecteront sur le serveur en simultané, ainsi il faut faire attention au moindre octet afin que le serveur n'ait aucun problème, j'ai ainsi appris qu'en informatique la notion d'infini n'existe pas, par exemple, une requête SQL doit toujours avoir une limite afin d'éviter les problèmes.

Cela m'a montré qu'un analyste programmeur n'arrête jamais d'apprendre, toute sa carrière il devrait se mettre à niveau, en effet l'évolution de l'informatique étant très rapide, un développeur ne peut se figer sur ses connaissances.

3.3 Bilan personnel

Dans un cadre personnel, ce stage fût une excellente expérience, cela m'a conforté dans mon choix de poursuivre mes études, en effet le stage était très intéressant, cependant je souhaite aller plus loin afin de prendre plus de décisions et de pouvoir évoluer dans ma carrière.

J'ai donc choisis d'aller en L3 afin de continuer vers un master.

Cela m'a également montré que j'étais capable de m'intégrer rapidement dans une équipe de développement, et que je pouvais ainsi poser ma pierre à l'édifice. Ce stage m'a donné confiance en moi, m'a montré que mes connaissances techniques actuelles pouvaient me permettre de développer beaucoup de chose, cependant mon avidité de connaissance me pousse à continuer.

Afin de continuer à faire évoluer le produit, j'ai eu la chance d'être employé en CDD au mois de juillet dans cette entreprise, cela me permettra je l'espère d'avoir une meilleure vision du future de l'application.

Ci-après mon cv avant et après le stage.

Antoine de Roquemaurel

23, avenue André Bousquairol

31 400 Toulouse

Tél. : 06 84 33 52 93

Email : antoine.roquemaurel@gmail.com

Né le 08 novembre 1991

DÉVELOPPEUR STAGIAIRE



EXPÉRIENCES

EXPÉRIENCES PROFESSIONNELLES

- Août 2009** **ÉLABORATION D'UN SITE INTERNET**, *Yverdon-les-bains, Suisse.*
Développement de toute l'architecture du site d'un artisan verrier, destiné à exposer ses pièces. Il est géré grâce à PHP, l'artisan peut donc l'administrer seul, comme il l'avait demandé.
2 semaines
▷ <http://valeriederoquemaurel.com>
- 2009 – 2012** **ANIMATION BÉNÉVOLE D'ENFANTS DE 6-11 ANS**, *Toulouse, France.*

PROJETS PERSONNELS

- 1er décembre 2011** **PARTICIPATION À LA NUIT DE L'INFORMATIQUE 2011**, *Toulouse, France.*
▷ <http://nuitdelinfo.com/nuitinfo11/>
- 2 décembre 2010** **PARTICIPATION À LA NUIT DE L'INFORMATIQUE 2010**, *Toulouse, France.*
Concours national d'informatique réunissant des étudiants de toute la France. Le concept est de développer une application web en une nuit.
▷ <http://nuitdelinfo.com/nuitinfo10/>
- 2006 – 2009** **DÉVELOPPEMENT D'UN SITE INTERNET.**
Dans le cadre d'un loisir, développement, maintenance et administration d'un site Internet sur le manga Bleach.
▷ <http://bleach.joohoo.fr>

FORMATIONS

- 2011 – 2012** **2ÈME ANNÉE DE DIPLÔME UNIVERSITAIRE DE TECHNOLOGIES INFORMATIQUE**,
Institut Universitaire de Technologies - Université Toulouse III, France.
- 2010** **BACCALAURÉAT SCIENCES ET TECHNIQUES DE L'INDUSTRIEL GÉNIE ÉLECTROTECHNIQUE**, *Mention assez bien, Lycée Saint Joseph Toulouse, France.*

LANGUES ET INFORMATIQUE

COMPÉTENCES TECHNIQUES

- Programmation** C, C++, Java.
- Web** XHTML, CSS, PHP.
- Versionnement** Git, Subversion.
- SGBD** MySQL, Oracle.
- IDE** Netbeans, Eclipse.
- Système** Linux, UNIX, Windows.
- Bureautique** LaTeX, Suite OpenOffice, Suite Microsoft Office.
- Certification** Certificat Informatique et Internet.

COMPÉTENCES EN LANGUE

- Anglais** Technique lu et écrit.

LOISIRS ET ACTIVITÉS

- Associatif** animateur Scouts et Guides de France
En formation de Brevet d'Aptitude à la Fonction d'Animateur.
- Culture** Lecture de science fiction (Asimov) et de fantastique (Tolkien).

Antoine de Roquemaurel

23, avenue André Bousquairol

31 400 Toulouse

Tél. : 06 84 33 52 93

Email : antoine.roquemaurel@gmail.com

Né le 08 novembre 1991

DÉVELOPPEUR



EXPÉRIENCES

EXPÉRIENCES PROFESSIONNELLES

- 2012** **REFONTE D'UN SYSTÈME DE TRADUCTION**, Memobox/2G Technologies, *Montrabé, France*.
STAGE Développement d'un nouveau système de traduction afin d'avoir une application web multi-lingue.
11 semaines J'ai ainsi développé un moteur de traduction en PHP Orienté Objet et une application permettant de traduire et rechercher dans la base de données de traduction, avec la technologie AIR de Adobe.
- Août 2009** **ÉLABORATION D'UN SITE INTERNET**, *Yverdon-les-bains, Suisse*.
2 semaines Développement de toute l'architecture du site d'un artisan verrier, destiné à exposer ses pièces. Il est développé en PHP, l'artisan peut donc l'administrer seul, comme il l'avait demandé.
▷ <http://valeriederoquemaurel.com>
- 2009 – 2012** **ANIMATION BÉNÉVOLE D'ENFANTS DE 6-11 ANS**, *Toulouse, France*.

PROJETS PERSONNELS

- 1er décembre 2011** **PARTICIPATION À LA NUIT DE L'INFORMATIQUE**, *Toulouse, France*.
Concours national d'informatique réunissant des étudiants de toute la France. Le concept est de développer une application web en une nuit.
2 décembre 2010 ▷ <http://nuitdelinfo.com/nuitinfo2010>
▷ <http://nuitdelinfo.com/nuitinfo2011>
- 2006 – 2009** **DÉVELOPPEMENT D'UN SITE INTERNET**.
Dans le cadre d'un loisir, développement, maintenance et administration d'un site Internet sur le manga Bleach.
▷ <http://bleach.joohoo.fr>

FORMATIONS

- 2011 – 2012** **2ÈME ANNÉE DE DIPLÔME UNIVERSITAIRE DE TECHNOLOGIES INFORMATIQUE**,
Institut Universitaire de Technologies - Université Toulouse III, France.
- 2010** **BACCALAURÉAT SCIENCES ET TECHNIQUES DE L'INDUSTRIEL GÉNIE ÉLECTROTECHNIQUE**, *Mention assez bien, Lycée Saint Joseph Toulouse, France*.

LANGUES ET INFORMATIQUE

COMPÉTENCES TECHNIQUES

- Programmation** C, C++, Java.
- Web** XHTML, CSS, PHP, JavaScript, Adobe AIR.
- Versionnement** Git, Subversion.
- SGBD** MySQL, Oracle.
- IDE** Netbeans, Eclipse.
- Système** Linux, UNIX, Windows.
- Bureautique** LaTeX, Suite OpenOffice, Suite Microsoft Office.
- Certification** Certificat Informatique et Internet.

COMPÉTENCES EN LANGUE

- Anglais** Technique lu et écrit.

LOISIRS ET ACTIVITÉS

- Associatif** animateur Scouts et Guides de France.