

Rapport de stage

Vincent Larchet

ISIA

Input	Rapport	Version	Date	Auteur(s)	Etat	Visa
644-221	-	1.0	18.07.02	VLA	Valid	

I. Table des matières

I.	Table des matières	2
II.	Journal des modifications au document	3
III.	Liste des documents de référence.....	3
IV.	Abréviations	3
1	Introduction	5
1.1	But de ce document	5
1.2	Structure du document.....	5
2	FastTicketting.....	6
2.1	Histoire.....	6
2.2	Améliorations	6
2.3	EBS.....	7
3	SecuTix™	8
3.1	Histoire.....	8
3.2	Brevet	8
3.3	IExpo02.....	9
3.4	Le futur.....	9
4	LEAF-Mobile	10
4.1	Histoire.....	10
4.2	Principes de LEAF	10
4.3	Développement.....	10
4.4	Gestion de projet.....	11
5	Conclusion	12

II. Journal des modifications au document

Nom du fichier	Version	Date	Remarques / Auteur
RapportTechnique.doc	0.1	20.06.02	Initial
rapportTechnique.doc	1.0	21.06.02	Final

III. Liste des documents de référence

IV. Abréviations

CORBA	Bus de communication entre services indépendants de l'implémentation des services
EBS	E-Business Server
EJB	Enterprise Java Bean
LEAF	Lean ELCA Architecture Framework
LEAF-Mobile	Adaptation de LEAF aux PDA
MFC	Microsoft Foundation Classes
PDA	Personal Digital Assistant (=palm)
RMI	Remote Method Invocation (=java RPC)
RPC	Remote Procedure Call
SMS	Short Messaging System

Remerciements

Je tiens d'abord à remercier tous les collaborateurs ELCA qui m'ont accompagné pendant toute la durée de ce stage.

Enfin, il va sans dire que je remercie aussi toute l'équipe de l'ISIA, à savoir Michel, Josianne, Jean-Charles et Sébastien pour leur accueil et leur soutien.

1 Introduction

1.1 But de ce document

Ce document est un rapport « technique » clôturant et résumant mon stage long ISIA.

Je vais y discuter de points techniques sur lesquels j'ai pu travailler mais aussi des aspects gestion de projets...

En premier lieu, il faut remarquer que mon travail a eut comme fil directeur les technologies mobiles et les PDA.

1.2 Structure du document

Je vais commencer par parler du projet FastTicketting, innovation permettant d'utiliser son téléphone portable comme un ticket.

Puis, je parlerai de ma participation au projet EBS, back-end d'un portail web qui utilise FastTicketting.

La troisième partie sera consacrée au projet SecuTix™, solution de ticketting permettant aux clients d'acheter leurs billets sur Internet et de les imprimer directement sur leur imprimante personnelle.

Et enfin, j'évoquerai ma participation dans LEAF-Mobile.

Notons que les projets SecuTix™ et LEAF-Mobile sont toujours en cours et que je continue à travailler dessus.

En réalité, le plan de ce document suit l'ordre chronologique dans lesquels j'ai abordé les différents sujets de travail.

2 FastTicketing

2.1 Histoire

FastTicketing est né avant mon arrivé chez ELCA par l'envie d'utiliser les nouvelles possibilités offertes par un téléphone portable. Le SMS venant d'arriver, on a pensé s'en servir pour envoyer des données à un client. L'idée est triviale mais ces données (un code et des données spécifiques à la prestation) sont considérées comme étant des droits d'accès à une prestation : ils venaient d'inventer le ticket par SMS.

Cette technique permet donc de vendre via Internet des tickets à des clients porteurs de téléphones portables, cela permet donc à n'importe quel prestataire d'augmenter son réseau de distribution tout en diminuant ses coûts car il s'affranchit de l'impression d'un billet, de sa vente dans des guichets (coût de location des locaux, du personnel...), des frais de port, etc.

Une fois un ticket émis, via un SMS, il faut encore, lors de l'arrivée du client sur le lieu de la prestation, contrôler ce billet afin d'autoriser, ou non, le client à entrer.

Si le contrôleur est dans un lieu fixe, alors une page web ou un GUI dans lequel on saisit le code fait l'affaire mais lors d'un contrôle mobile (dans un bus, train...), cette solution ne s'applique pas. L'idée était d'utiliser, encore une fois, la technologie SMS : le contrôleur envoie un SMS contenant le code du billet à un numéro de téléphone spécial et reçoit un autre SMS en retour lui indiquant si le billet est valide ou non. Ce système, bien qu'ingénieux, possède néanmoins quelques inconvénients :

- Le contrôleur doit saisir un code, ce qui est assez long et est sujet à de nombreuses fautes de frappe (essayer de saisir sur les touches minuscules d'un téléphone portable un code alors que l'on est debout dans un train qui bouge est un bon exercice de concentration !!!)
- Si le contrôleur se trouve dans une zone sans couverture GSM, alors le contrôle ne fonctionne pas

2.2 Améliorations

La première tâche de mon stage a été d'optimiser le contrôle de ces billets.

Nous avons décidé d'utiliser un PDA pour le contrôleur. Il apporte les avantages suivants :

- Les touches à l'écran sont bien plus grandes, et il est facile de corriger ou empêcher certaines fautes de frappes (caractères alphanumériques à la place de chiffres...)
- Il est possible d'embarquer du code dans le PDA pour effectuer des vérifications sur le code saisi sans avoir de connexion à un serveur,
- Le résultat du contrôle peut être beaucoup plus détaillé et beaucoup plus facile à lire grâce au grand écran couleur des PDA modernes,

- Le contrôle s'en retrouve plus rapide car le contrôleur à moins d'opérations manuelles à effectuer

J'ai donc écrit une application sur un PDA (iPaq 3630) permettant de faire tous ces points.

Ensuite, nous avons voulu améliorer encore le système et nous avons pensé à utiliser des code-barres représentant le code du billet (le même que celui affiché sur le SMS) et avec un lecteur de code barre sur le PDA, le contrôle devenait très rapide et très fiable car plus aucune opération manuelle de la part du contrôleur n'est plus alors nécessaire.

Durant cette période, j'ai aussi commencé à participer au « groupe mobile » chez ELCA. Ce groupe d'intérêt rassemble les gens s'intéressant aux technologies mobiles et permet de mettre en commun nos différentes connaissances et idées afin de créer de nouvelles applications et ainsi pouvoir proposer à nos clients des projets s'appuyant sur les toutes dernières technologies.

2.3 EBS

FastTicketting ayant eut un certain succès, une grande entreprise de transport a décidé, lors de la refonte de son portail web, d'utiliser FastTicketting afin d'élargir son offre.

J'ai participé à ce projet sur de nombreux points :

- Implémentation de la gateway SMS pour envoyer les billets FastTicketting
- Gestion de configuration en réalisant des scripts de synchronisation de repository CVS car le projet était réparti entre des développeurs en Suisse et au Vietnam
- Implémentation d'un framework de tests de non-régression automatiques
- Et de nombreuses classes d'outil pour envoyer des mails, imprimer des écrans Swing

3 SecuTix™

3.1 Histoire

Après les recherches effectuées en vue d'améliorer le contrôle des billets FastTicketing, nous avons essayé d'élargir une nouvelle fois les canaux de distribution des billets.

Depuis quelques temps, nous avons pensé à utiliser des code barres comme support pour un billet, nous avons donc essayé de faire quelque chose dans le même style : distribuer sur Internet des billets que les clients pourraient imprimer eux-mêmes sur leur propre imprimante.

Nous avons en plus comme objectif que le contrôle de ce billet soit le plus simple, le plus rapide et le plus robuste.

Etant à Lausanne, nous avons pris contact avec un laboratoire de l'EPFL qui a participé à l'élaboration des billets de banque suisse et qui est spécialisé dans la protection de données sur papier.

Après avoir étudié leurs techniques nous avons imaginé une solution de ticketing en modifiant un de leurs procédés et en l'utilisant à nos fins.

Cette solution consiste en quelques mots à utiliser une méthode de tramage particulière permettant de tramer une image avec du texte. Cela permet d'ajouter de l'information textuelle sur une image sans en modifier trop l'aspect.

Ceci combiné avec la personnalisation des billets en fonction du client et de l'événement permet d'atteindre un bon niveau de sécurité, avec uniquement un contrôle visuel (il est donc possible de s'affranchir du code barre).

3.2 Brevet

Cette solution nous apparaissant comme innovatrice, nous avons choisi de déposer un brevet sur la technique que nous avons mise en place. Tout d'abord, j'ai écrit une première version d'un brevet pour protéger notre solution. Après deux mois de labeur acharné sur l'écriture de ce brevet, le résultat n'étant toujours pas pleinement satisfaisant, nous avons décidé de faire appel à un avocat spécialisé dans la rédaction des brevets. Notre ébauche de brevet, d'autres explications et quelques documents ont permis à cet avocat d'écrire un brevet robuste au niveau légal.

Le brevet résultant a été déposé aux Etats Unis le 27 janvier avec possibilité d'extension au reste du monde dans les 30 mois. Il n'est bien évidemment pas encore accepté et les discussions avec l'USPTO sont en cours.

Après quelques mois de recherche, nous avons donc pu mettre en place une solution assurant un bon niveau de sécurité (non-falsification) avec un billet imprimé directement chez le client et dont un contrôle uniquement visuel est possible et suffisant. Afin de s'assurer un maximum de sécurité, nous avons couplé le contrôle visuel avec un système de vérification électronique basé sur un code barre dans

lequel les données relatives au client, à la prestation... sont cryptés, puis décryptés par l'appareil du contrôleur.

Pour plus d'information, cf www.secutix.com

3.3 IExpo02

Internet Expo est la plus grande manifestation informatique de Suisse, rassemblant près de 500 exposants pour plus de 40 000 visiteurs.

ELCA y possédant un stand, nous avons profité de cette occasion pour faire une conférence d'annonce, des revues de presses... afin de faire connaître notre innovation à d'éventuels partenaires ou entreprises susceptibles d'être intéressés.

Nous avons en plus mis en place une démonstration interactive de notre solution où les personnes passant à notre stand pouvaient s'arrêter et en quelques clics recevoir un billet personnalisé fictif de concert, train... où apparaissait leur nom, photo, la date...

3.4 Le futur

La solution SecuTix™ connaît un succès non négligeable et est en train d'être appliquée à deux cas réels. Les premiers billets réellement vendus et achetés par des clients non fictifs devraient voir le jour vers le mois d'octobre.

4 LEAF-Mobile

4.1 Histoire

Après avoir beaucoup développé sur des PDA, nous nous sommes rendus compte qu'une bonne partie des problèmes soulevés à chaque fois étaient les mêmes.

Nous avons commencé par chercher un produit répondant aux besoins de réutilisabilité que nous avions mais rien ne nous a paru satisfaisant. Or, une des spécialités d'ELCA est la création de framework de développement pour économiser du temps de développement et minimiser les bugs, nous avons donc pris contact avec la division Architecture. Après discussion, nous sommes parvenus à la conclusion que le modèle de composants LEAF qui est développé pour les environnements distribués Java/EJB correspondait bien à nos attentes. De plus, il nous semblait intéressant de pouvoir, à partir d'un PDA, nous interfacer avec un serveur java/EJB.

Nous avons donc choisit d'écrire nous même un framework, basé sur les principes de LEAF.

4.2 Principes de LEAF

LEAF comporte de nombreux principes architecturalement très intéressants, dont :

- « Service oriented » : cela veut dire que l'architecture des composants n'est pas orientée objet mais service. Cela donne une séparation des données, du traitement de ces données et de leur stockage.
- « location transparency » : cela veut dire qu'un composant logiciel en utilisant un autre de façon transparente sans se soucier de savoir s'il est écrit dans le même langage, s'il est sur la même machine ou distant... Cela veut aussi dire, à l'opposé, qu'un composant sera écrit d'une façon unique qu'il soit destiné à être utilisé en local ou en distant, via CORBA, RMI... : il devient indépendant de la technologie qui l'utilisera.
- « LEAF layer » : c'est une couche qui se rajoute automatiquement avant les services et qui permet de rajouter dynamiquement ou lors de la configuration des intercepteurs sur les appels de services.
- « coding support » : permet d'ajouter des pré-post conditions sur les méthodes (design by contract), gère les traces configurables...

4.3 Développement

Ce framework est écrit en C++/MFC. Il fonctionne sur tous les PDA basés sur le système d'exploitation windows CE.

Le développement a commencé début mars, nous sommes 3 personnes à temps partiel et nous avons déjà un noyau fonctionnel sur lequel deux applications sont en train de se bâtir.

Nous avons réussi à implémenter en C++ tous les mêmes mécanismes que dans LEAF, i.e., l'instanciation dynamique de classes, la « location transparency » grâce à la génération automatique de stub/skeleton et l'utilisation de nom logique pour désigner les services, couche Leaf (invokers, intercepteur...), intégration du système de build automatique, déploiement automatique...

4.4 Gestion de projet

Ce projet est une grande première pour moi car ayant une certaine expérience dans le développement sur PDA, et ayant une bonne connaissance de l'environnement LEAF en java/EJB, j'ai été choisi pour prendre en main ce projet.

C'est donc moi qui donne les grandes orientations de LEAF-Mobile, qui distribue le travail...



5 Conclusion

En conclusion, ce stage, de part sa diversité tant d'un point de vue technique que des méthodes de travail ou des relations humaines, m'a enchanté. J'ai aussi beaucoup apprécié la confiance que l'équipe dirigeante n'a pas hésité à placer en moi et me confiant ou me faisant participer à des projets innovants et stratégiques.

Toutes ces raisons font que j'ai choisi de continuer à travailler chez ELCA Informatique SA.