

MEMOIRE D'ACTIVITE PROFESSIONNELLE	
Nom de l'auditeur : INTHAVONG	Nom de l'entreprise : IER
Fonction : Analyste programmeur	Adresse : 3, rue Salomon de Rothschild 92 156 Suresnes - Cedex
Dur�e du projet : 7 mois	
Dates : Mars � Septembre 2002	Service : Etudes Terminaux
Libell� : D�veloppement du logiciel de lecteur d�di� au contr�le d'acc�s � bord d'avions, bateaux et trains avec reconnaissance d'empreintes digitales.	
R�sum� : La soci�t� IER est sp�cialis�e dans la conception, l'industrialisation et la commercialisation de terminaux et syst�mes sp�cifiques d'impression de billetterie des grands r�seaux de transport a�rien, maritime, ferroviaire et d'agences de voyages. L'�v�nement du 11 Septembre a conduit les compagnies a�riennes et a�roports � r��valuer toutes les �tapes du trajet du passager. Le d�fi auquel l'industrie est confront�e aujourd'hui est d'accro�tre la s�curit� dans les a�roports. IER a propos� � ses clients, le lecteur sp�cialis� permettant le contr�le d'acc�s � bord d'avions, de bateaux et de trains avec reconnaissance d'empreintes digitales et la possibilit� de lire les billets avec code � barres. J'ai �t� appel� � int�grer l'�quipe de d�veloppement de ce lecteur. Ma mission a consist� dans un premier temps, � analyser les besoins clients exprim�s dans le cahier des charges puis � r�diger les sp�cifications et enfin la conception et le d�veloppement du logiciel du lecteur.	
Mots cl�s : Lecteur sp�cialis�, d�veloppement logiciel.	
Date de la Soutenance :	
Nombre de pages :	Notes du destinataire :

SOMMAIRE

1. INTRODUCTION.....	4
2. PARCOURS PROFESSIONNEL	5
3. PRESENTATION DE L'ENTREPRISE	9
3.1. COMPETENCES ET SAVOIR-FAIRE	9
3.2. ETUDES ET DEVELOPPEMENTS.....	10
3.3. SITE DE PRODUCTION	10
3.4. DOMAINE DE COMPETENCES	10
3.4.1. <i>Domaine du Transport de passagers.....</i>	<i>10</i>
3.4.2. <i>Domaine des bornes libre-service et de l'impression spécifique.....</i>	<i>11</i>
3.4.3. <i>Domaine d'identification par radiofréquence.....</i>	<i>11</i>
3.5 PRESENTATION DU SERVICE ETUDES TERMINAUX	14
3.6. CHIFFRES D'AFFAIRES	15
4. ACTIVITES PROFESSIONNELLES	16
4.1. FONCTIONS ET MISSIONS	17
4.2. TACHES	20
4.3. CRITERES DE PERFORMANCE.....	21
4.4. RELATIONS INTERNES ET EXTERNES.....	21
4.4.1. <i>Relations internes</i>	<i>21</i>
4.4.2. <i>Relations externes</i>	<i>22</i>
5. PROJET REALISE.....	23
5.1. PRESENTATION DU LECTEUR IER 627BC	23
5.2. ARCHITECTURE MATERIELLE DU LECTEUR.....	25
5.2.1. <i>Carte principale.....</i>	<i>26</i>
5.2.2. <i>Carte d'interface.....</i>	<i>26</i>
5.3. LES CONTRAINTES DE DEVELOPPEMENT.....	28
5.4. LE CYCLE DE DEVELOPPEMENT	28
5.4.1 <i>Avantages et inconvénients du cycle en V</i>	<i>29</i>
5.5. PROCESSUS DE DEVELOPPEMENT	33
5.5.1. <i>Définition du protocole d'échange de la carte interface</i>	<i>33</i>
5.6. DEGRE D'AVANCEMENT	38
5.7. BILAN TECHNIQUE	39
5.8. OPTIMISATION MISE EN OEUVRE.....	40
5.8.1. <i>Optimisation documentaire</i>	<i>40</i>
5.8.2. <i>Optimisation au niveau du développement</i>	<i>41</i>

6. COMPOSANTES DU METIER D'INGENIEUR-MAITRE	41
6.1. FONCTIONS TECHNIQUES.....	42
6.2 FONCTIONS DE GESTION ET D'ENCADREMENT	42
6.3. FONCTION DE TRANSMISSION.....	43
6.4. FONCTION FINANCIERE	43
 7. THEME D'ETUDE : LA PROBLEMATIQUE DU LOGICIEL.....	 44
7.1. LE GENIE LOGICIEL.....	44
7.2. LA CRISE DU LOGICIEL.....	44
7.3. LES REPONSES A LA CRISE	45
7.3.1 <i>La maîtrise du processus de développement</i>	45
7.3.2. <i>Développement de méthodes structurées</i>	46
7.3.3. <i>Le paradigme objet</i>	46
7.3.4. <i>L'approche composants</i>	47
 8. CONCLUSION TECHNIQUE.....	 49
 9. CONCLUSION GENERALE.....	 50

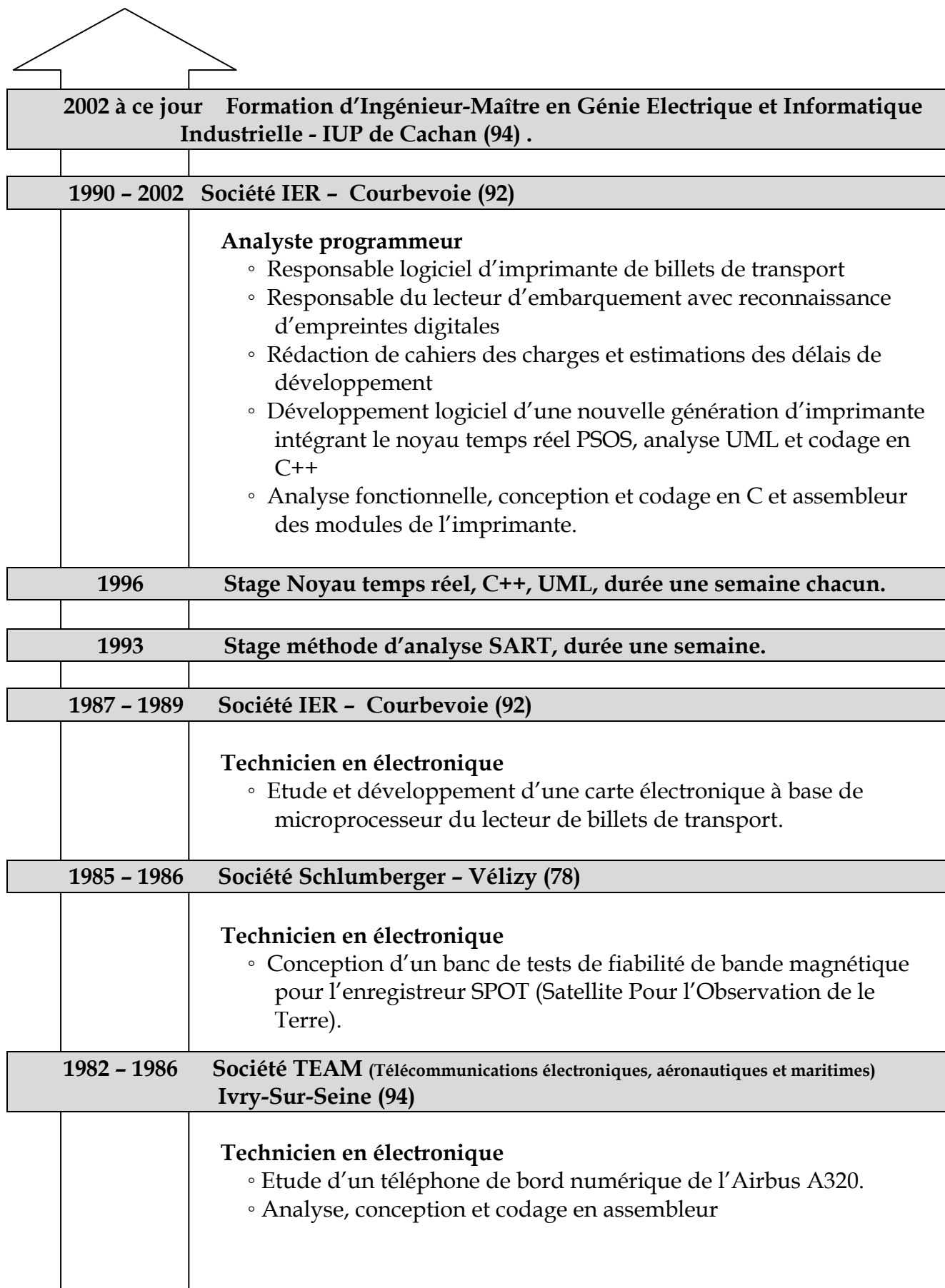
1. Introduction

Ce mémoire retrace la mission la plus récente que j'ai effectuée au cours de mon activité professionnelle depuis 1987 dans la société IER qui est spécialisée dans la conception, l'industrialisation et la commercialisation de terminaux et systèmes spécifiques d'impression.

La mission sur laquelle j'ai été affectée concerne la mise en oeuvre du lecteur spécialisée qui avait pour but de contrôler l'accès à bord d'avions, de bateaux ou de trains. Ce lecteur sera doté d'une fonction innovatrice qui est la reconnaissance d'empreintes digitales permettant d'accroître la sécurité dans les transports.

Ce mémoire présentera les différentes missions qui m'ont été confiées durant mon activité professionnelle depuis mon intégration au sein de la société. Ensuite, j'approfondirai mon activité principale au sein du projet qui concerne le développement logiciel du lecteur. Il sera complété par la mise en évidence des différentes formes d'optimisations que j'ai mises en oeuvre pour rendre le logiciel fonctionnel dans un délai très court.

2. Parcours professionnel



1982

**DUT Génie Electrique et Informatique Industrielle
Option Electrotechnique - IUT de Cachan (94)**

Mon parcours professionnel débute après l'obtention du DUT Génie Electrique et Informatique Industrielle, option Electrotechnique en 1982 à l'Institut Universitaire de Cachan - Université Paris XI Orsay.

J'ai été embauché en CDI par la société TEAM de 250 salariés (Télécommunications Electroniques Aéronautiques et Maritimes) qui concevait et fabriquait des téléphones de bord pour l'aviation.

J'ai intégré d'abord le service après-vente où ma mission a été le dépannage de cartes électroniques du téléphone de bord. Après six mois, j'ai trouvé que ce travail était monotone et ne correspondait pas à mon inspiration. J'ai donc fait une demande pour intégrer le service R&D et qui a été acceptée au bout de quelques mois.

Mon attirance par l'informatique en générale, m'a conduit à suivre en parallèle à mon activité professionnelle, les cours du soir au CNAM dans le cycle électronique afin d'étendre mon champ de connaissances et dans l'espoir d'évoluer vers le poste plus valorisant. Les cours suivis en 1983 à 1985 étaient les mathématiques générales, l'électronique et l'architecture des systèmes.

Ma principale mission dans le service R&D était la conception et la réalisation du téléphone de bord numérique pour l'Airbus A320. Ce projet m'a permis de mettre en pratique toutes mes connaissances en électronique numérique et en programmation assembleur que j'ai acquises durant mes études.

J'ai eu l'opportunité de concevoir et réaliser le projet entièrement dont voici les différentes phases :

- conception de schémas électroniques ;
- choix et l'achat des composants ;
- lancement de circuits imprimés auprès des sous-traitant ;
- réalisation du programme en assembleur ;
- mise au point du programme et l'intégration ;
- tests de performance et d'endurance de l'ensemble.

Après une expérience sérieuse dans le domaine électronique et informatique, j'ai demandé une évolution de carrière qui m'a malheureusement été refusée. J'ai donc prospecté sur le marché du travail et j'ai été embauché en 1986 par la société Schlumberger. En tant que Technicien en Electronique, j'avais pour mission de concevoir un banc de test de bande magnétique pour l'enregistreur SPOT (Satellite Pour l'Observation de le Terre) permettant le test de fiabilité des bandes magnétiques. L'électronique du banc ne mettant en application que de la logique combinatoire, m'a peu attiré et l'évolution de carrière au sein du service n'était pas certaine. J'ai décidé donc de mettre fin à mon contrat.

Ensuite, j'ai été embauché, en CDI par la société IER de 250 salariés qui concevait et fabriquait des terminaux et systèmes spécifiques d'impression, d'abord en tant que technicien en électronique, j'ai développé la carte électronique à base de microprocesseur pour le lecteur de billets de transport ATB (Ticket d'accès à bord).

A l'époque, l'informatique embarquée commençait à prendre de l'envol dans le monde industriel. J'ai ressenti que l'informatique allait être le métier de l'avenir ; j'ai eu l'opportunité d'intégrer le nouveau service Logiciel, lorsque le service R&D auquel j'ai appartenu, a été scindé en trois entités : service Mécanique, service Electronique et service Logiciel. En tant qu'analyste programmeur depuis 1990, j'ai développé des applications spécifiques en assembleur pour des lecteurs, puis je me suis adapté aux langages C, C++ et UML suite à des formations financées par la société. Ensuite, j'ai été responsable du « logiciel de base » d'imprimantes professionnelles IER et j'ai encadré quelques prestataires, de nouveaux ingénieurs et un programmeur pour les développements des applications spécifiques.

3. Présentation de l'entreprise

Depuis sa création en 1962, la société IER est spécialisée dans la conception, l'industrialisation et la commercialisation de terminaux et systèmes spécifiques d'impression se situant à l'avant garde des technologies.

IER ne cesse de rechercher et mettre en œuvre des technologies innovantes pour garantir la pérennité des investissements de ses clients.

Depuis plus de 15 ans, IER a développé un réseau de filiales à l'étranger, Etats-Unis, puis Singapour, Royaume-Uni, Chine, Japon, Dubaï, Suède, Espagne, Belgique qui assure une présence commerciale et technique auprès des clients du monde entier.

Chiffres clés :

- Effectifs : 1 000 collaborateurs dans le monde
- Base installée : 280 000 terminaux et systèmes
- Production : capacité de 30 000 terminaux / an

Les atouts :

- Une forte implantation nationale et internationale
- Une politique qualité rigoureuse
- Des bureaux d'études comprenant plus de 100 personnes

3.1. Compétences et savoir-faire

Fort de près de 40 années d'expérience, IER maîtrise complètement la chaîne industrielle (conception, fabrication, intégration et maintenance) et dispose d'un savoir-faire unique pour développer et mettre en œuvre des solutions sur mesure fiables et innovantes.

IER dispose ainsi d'un ensemble de compétences avancées dans les domaines :

- études, design ,
- mécanique, électronique, informatique,
- industrialisation, intégration, adaptation,
- installation, formation,
- maintenance.

Cette position reconnue de leader mondial, grâce à la présence sur le terrain, à la politique de Qualité ainsi qu'au savoir-faire industriel et technique acquis depuis 40 ans.

3.2. Etudes et développements

Chaque année, des investissements importants sont consacrés à la recherche et au développement de solutions innovantes basées sur les technologies les plus avancées. Doté de l'un des plus importants bureaux d'études d'imprimantes, de terminaux spécialisés et de bornes interactives du marché, IER s'appuie sur une politique de Qualité totale, depuis la conception jusqu'à la mise en œuvre opérationnelle.

Une équipe de près de 100 personnes assure chaque jour :

- la mise en place d'un partenariat adapté entre votre société et notre bureau d'études ;
- des délais de conception et de fabrication raccourcis grâce à l'utilisation de la modélisation mécanique en 3 D et de la simulation électronique ;
- la prise en compte des contraintes d'utilisation et de maintenance dès la phase de conception.

3.3. Site de production

Implanté depuis près de 20 ans au cœur de la Franche Comté, région réputée pour la qualité et les compétences de sa main-d'œuvre en micro-mécanique et micro-électronique, IER dispose d'un site de fabrication sur mesure.

Cette usine de Besançon, sans cesse modernisée, permet une capacité annuelle de production de près de 30 000 systèmes.

Un concept d'atelier flexible, la rigueur des procédures de Qualité industrielle et un système intégré de Gestion de Production Assistée par Ordinateur permettent de répondre aux impératifs de livraisons de ses clients.

3.4. Domaine de compétences

3.4.1. Domaine du Transport de passagers

IER est leader mondial pour l'équipement de terminaux de billetterie, enregistrement et accès à bord des grands réseaux de transport aériens, ferroviaires et maritimes, des réseaux d'agences de voyages et des aéroports.

Plus 250 000 imprimantes et lecteurs ATB (Ticket d'accès à bord) ainsi que des imprimantes d'étiquettes à bagages sont installés dans plus de 800 aéroports et près de 200 compagnies aériennes aussi dans près de 15000 d'agences de voyages dans 80 pays et dans 7500 gares.

3.4.2. Domaine des bornes libre-service et de l'impression spécifique

IER conçoit et fabrique des bornes interactives, standards ou spécifiques tel que :

- la borne d'information pour la présentation multimédia des produits et services ou promotion du site Internet de l'entreprise ;
- la borne d'aide à la vente de tickets d'entrée dans les musées, les stades, les cinémas, les salles de spectacle, les stations de sports d'hiver;
- la borne d'édition de documents administratifs comme les certificats de non gage ou autres formulaires, bons de commande et fiches produit ;
- la borne Libre-Service, cette borne permet l'achat en libre-service, dans des lieux publics de cartes téléphoniques, d'accès à Internet, de recharges pour téléphones mobiles ;
- borne d'affranchissement du courrier dans les postes.

3.4.3. Domaine d'identification par radiofréquence

Depuis déjà plusieurs années IER a développé une expertise dans la mise en œuvre de solutions d'identification intégrant des étiquettes radiofréquence pour répondre aux applications de traçabilité, là où le code à barres a atteint ses limites.

Une équipe d'ingénieurs et de techniciens dédiée à la technologie sans contact étudie et conçoit des solutions basées sur la radiofréquence et propose des étiquettes radio standards du marché, ou bien, dans le cas d'applications spécifiques, des étiquettes

packagées sur-mesure.

IER comptabilise à ce jour de nombreuses applications fonctionnelles dans les domaines d'Identifications des produits, des processus industriels:

- identification et suivi de plusieurs centaines de milliers d'arbres,
- identification et gestion de la maintenance du mobilier urbain et de panneaux d'affichage,
- identification et traçabilité des cylindres de gaz médicaux,
- suivi de containers et de bacs plastiques,
- traçabilité sur process de fabrication de postes de télévision,
- laverie industrielle.
- système de contrôle d'accès et de flux de passage

Les activités commerciales d'IER sont organisées comme montre la figure suivante.

Groupe IER
Direction opérationnelle

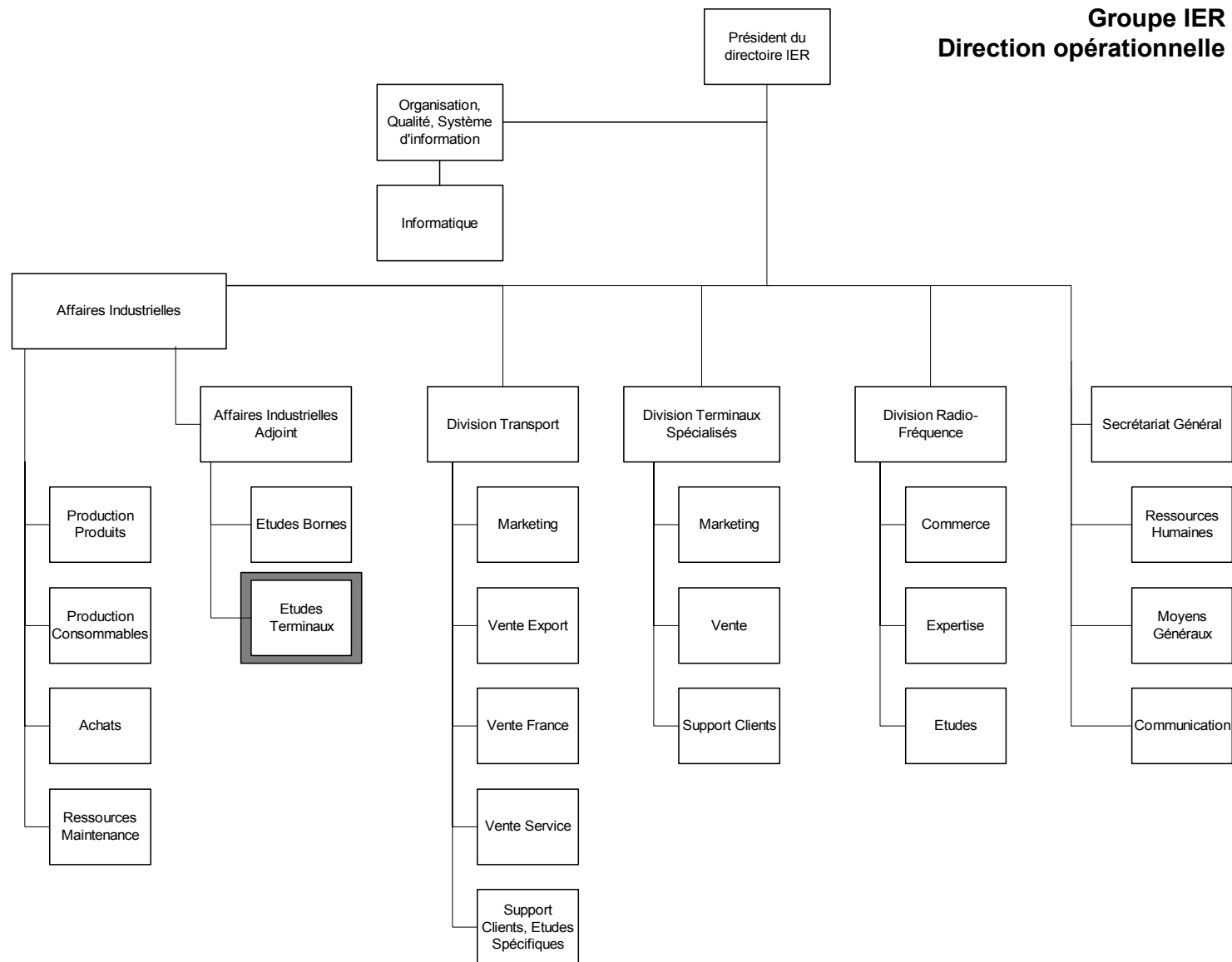


Fig.1. Organisation de la société IER

3.5 Présentation du service Etudes Terminaux

Le service Etudes Terminaux a en charge tous les projets concernant le domaine des transports de passagers. Il conçoit et développe principalement des imprimantes et lecteurs ATB, des imprimantes d'étiquettes à bagages. Cette unité de recherche et développement travaille en interaction avec les services Electronique, Mécanique, Logiciel... de façon à ce que le développement d'un produit profite à toute l'unité. La figure ci-après décrit l'organigramme de l'unité.

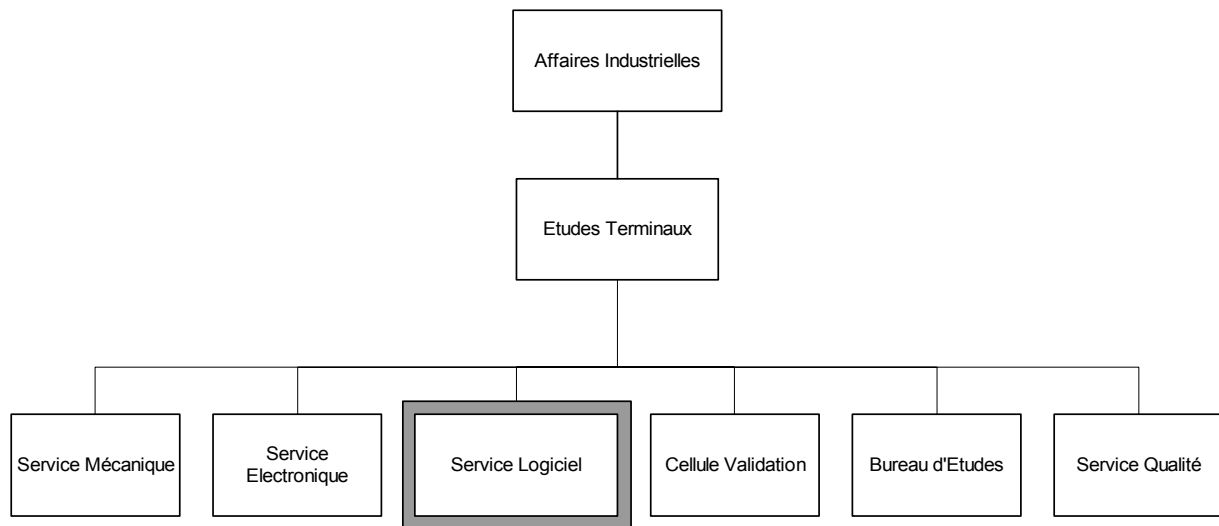
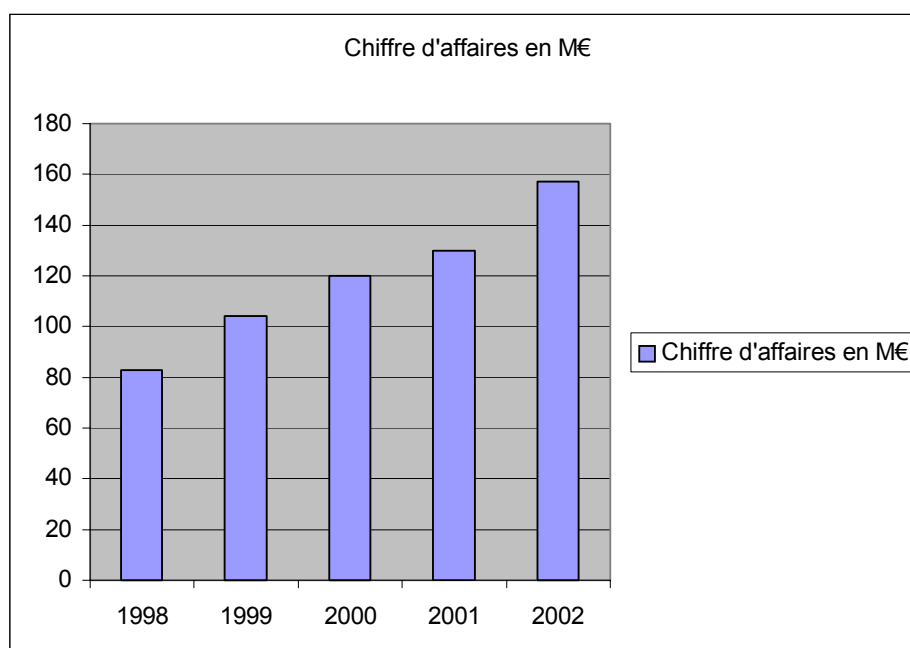


Fig. 2. Organigramme du service Etudes Terminaux

3.6. Chiffres d'affaires

IER est une société anonyme au capital de 12,5 M€. La figure ci-dessous montre que l'évolution du chiffre d'affaires est en progression constante depuis 1998.



4. Activités professionnelles

Mon expérience en développement logiciel, en mécanique et mes connaissances en électronique m'ont permis d'intégrer le service logiciel en 1988 (voir Fig. 4).

Les produits que j'ai été amené à développer et à faire évoluer, par ordre chronologique, sont :

- le lecteur permettant le contrôle d'accès à bord d'avions lors de l'embarquement ;
- l'imprimante professionnelle destinée à l'édition des coupons de voyages ;
- l'imprimante professionnelle nouvelle génération ;
- le lecteur destiné au contrôle d'accès à bord avec reconnaissance d'empreintes digitales.

L'expérience que j'ai acquise au fil des années, m'a permis d'approfondir mes compétences techniques, de perfectionner ma capacité à estimer le délai de développement logiciel, d'étendre ma capacité de rédaction des documents concernant les logiciels spécifiques et de me former aux techniques d'encadrement.

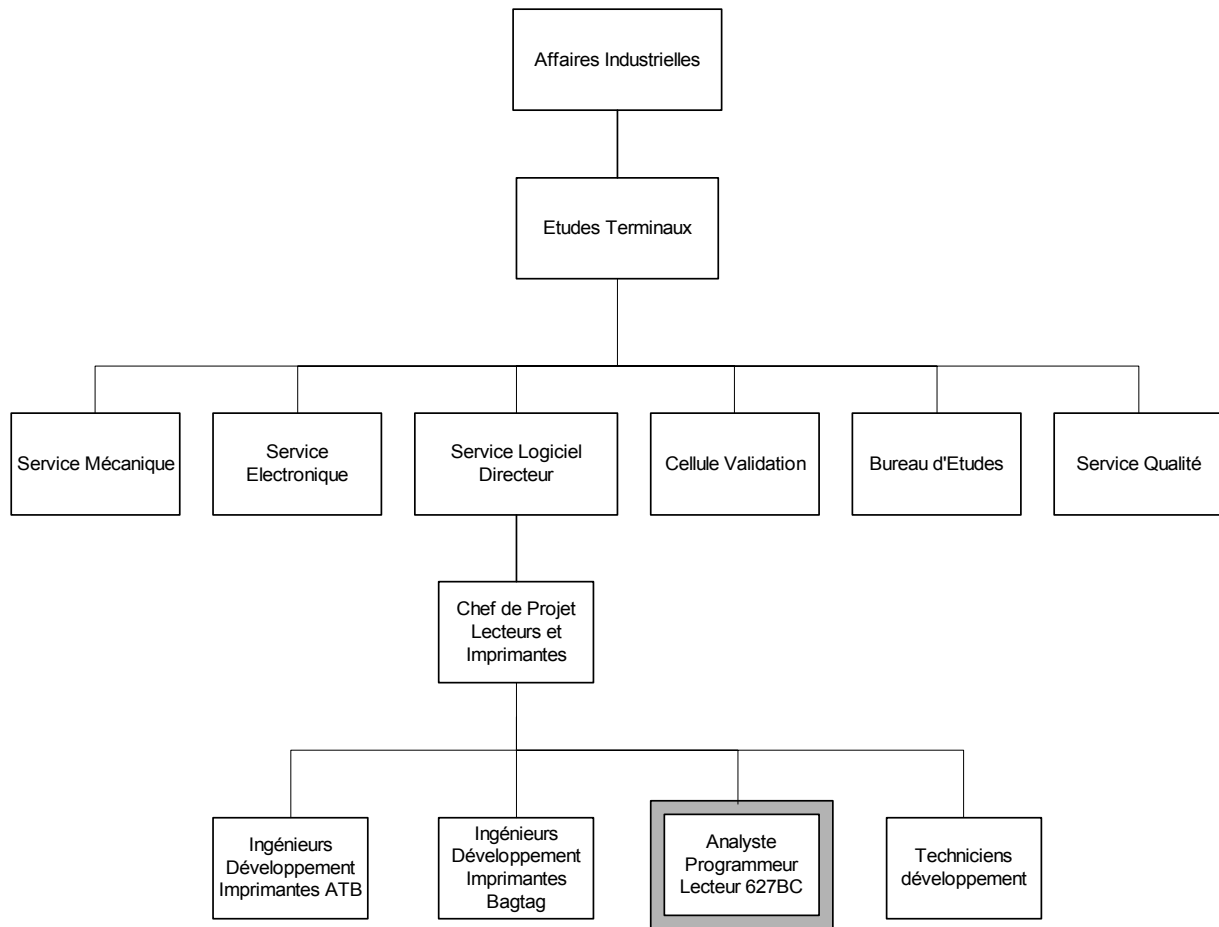


Fig. 4 - Localisation hiérarchique du poste

4.1. Fonctions et Missions

La division du service Etudes Terminaux est organisé en trois entités, Mécanique, Electronique et Logiciel, et mon intérêt pour l'informatique industrielle, m'ont conduit naturellement vers le service logiciel. J'ai intégré l'équipe pour développer le tout premier lecteur IER 617.

En tant qu'analyste programmeur, j'ai pour mission:

- d'analyser, concevoir et réaliser le logiciel de « base » (en C et Assembleur 80188),
- de développer des logiciels spécifiques pour les compagnies aériennes,

- d'améliorer les performances,
- de corriger les anomalies.

Quelques temps après, j'ai été promu responsable du « logiciel de base » du lecteur et j'encadrais deux prestataires pour les développements spécifiques.

En 1994, face à la demande croissante des clients et de concurrents très actifs, le français Dassault, l'américain Genicom et le japonais Datamax notamment, IER se lançait dans le développement d'une nouvelle imprimante professionnelle IER 557. Celle-ci devait remplacer l'IER 457 qui était le fleuron de la société depuis 5 ans. La nouvelle machine devait être plus performante, plus modulaire, moins encombrante, plus maintenable et surtout moins chère. J'ai été appelé à intégrer une nouvelle équipe de développement logiciel pour :

- migrer une partie du logiciel 457 (Assembleur 80188) vers la 557 (Assembleur 80186);
- analyser et concevoir trois logiciels des modules de l'imprimante;
- analyser et développer plusieurs parties du logiciel de l'imprimante.

En 1995 j'étais responsable « logiciel de base » de l'imprimante du fait que les ingénieurs de l'équipe quittaient ce projet pour développer une nouvelle imprimante l'IER567. En tant que responsable, j'encadrais quelques prestataires, de nouveaux ingénieurs et un programmeur pour les développements des applications spécifiques (~ 200 applications).

En 1998, j'avais suivi trois stages en UML, en C++ et noyau temps réel, avant de rejoindre l'équipe de développement IER 567, je participais à l'analyse et la conception des parties :

- impression de fontes Windows,
- impression de code à barres,
- chargeur client qui est un programme indépendant permettant de mettre à jour un programme client par téléchargement,

et je participais aussi à l'amélioration des performances et à la correction des anomalies.

Après l'événement du 11 Septembre 2001, les compagnies aériennes voulaient augmenter la sécurité à l'embarquement. IER avait développé le lecteur IER627BC permettant le contrôle d'accès à bord d'avion avec reconnaissance d'empreintes

digitales. Je suis donc jusqu'à ce jour, responsable logiciel de ce lecteur pour développer les applications spécifiques.

4.2. Tâches

Mes principales tâches consistait à :

- rédiger des spécifications logicielles relatifs aux imprimantes et aux lecteurs de «base »,
- assurer la formation sur l'architecture du logiciel aux nouveaux arrivants dans le projet,
- assurer l'archivage hebdomadaire des projets courants et mensuel de tous les projets,
- proposer des solutions techniques aux développeurs lorsque les idées soumises sont peu fiables ou inefficaces,
- estimer tous les développements spécifiques demandés par le service commercial.
- rendre compte à mon responsable direct l'état d'avancement de mon projet et les projets d'autres développeurs.

Un « programme de base », que ce soit celui de l'imprimante ou du lecteur contient un nombre impressionnant de fichiers compris entre 90 et 150 et le nombre d'applications spécifiques dérivant cette base représentent environ 200. Vue l'ampleur du projet, il était indispensable de mettre en place des règles de travail qui seraient applicables à tous développeurs d'applications:

- uniformisation des entêtes des fichiers, en faisant apparaître
 - l'issu du fichier,
 - le nom de l'auteur,
 - la date de modification,
 - le résumé ou le but de la modification;
- instauration des règles pour nommer les fonctions et variables globales;
- instauration du fichier lisezmoi.txt contenant le résumé du cahier des charges, le nom de l'auteur, la liste des fichiers modifiés et les fonctions modifiées, la durée réelle du développement. Cette durée me permettrait d'estimer avec exactitude le délai des développements futurs ;
- gestion des versions des fichiers et applications, dans le but de faciliter la comparaison de deux versions ou le retour en arrière des applications;

- rédaction des fiches de tests et création des macros associés pour les toutes les applications. Ces macros étaient très utiles, voire indispensables pour valider la non régression d'un programme lorsqu'il y a une évolution de version et que le développeur qui était à l'origine de la version antérieure n'était plus présent.

4.3. Critères de performance

Création des différents outils pour automatiser les tâches répétitives dans le développement du logiciel et le débogage (recherche de variables globales, recherche de fonctions, etc) Cela m'a permis de gagner 20% en performance ainsi que pour l'équipe.

Création outils permettant de générer automatiquement les étiquettes de disquette ou de EPROM lors de la diffusion de programme, évitant ainsi les erreurs de frappe et augmenter de quelques pour cent l'efficacité et d'avoir une uniformisation des étiquettes.

Respect d'efficacité : Tous développeurs devaient informer « l'homme de base » lorsqu'ils découvraient un dysfonctionnement relatif au programme de base, afin que celui-ci puisse corriger au plutôt et que les applications courantes et futurs en bénéficiaient. Sinon, les mises à jour risquaient de coûter cher une fois que les machines seront installées chez le client.

Respect des délais : Tous développements risquant de dépasser le délai fixé devaient être signalés au responsable hiérarchique afin qu'il puisse prendre les mesures nécessaires auprès du service commercial.

4.4. Relations internes et externes

4.4.1. Relations internes

Responsable hiérarchique direct : Mon responsable hiérarchique définissait mes objectifs. Je rapportais régulièrement l'état d'avancement de mon projet et ceux d'autres développeurs. S'il y avait des problèmes d'incompréhension du cahier des charges, c'est lui qui faisait l'interface avec le technico-commercial ou le client.

Cellule validation : La cellule validation était le représentant du client, elle testait toutes les applications développées par le service Logiciel avant la diffusion. Un rapport d'anomalies était dressé en fin de test et communiqué à mon responsable hiérarchique qui décidait de la priorité de corrections selon la gravité et le temps disponible.

Service électronique : Le service électronique développait toute l'électronique des machines. Il nous fournissait toutes les spécifications fonctionnelles et assurait le support technique des produits pendant la phase de développement .

Service mécanique : Ce service était sollicité lorsqu'il y avait des problèmes mécaniques sur les machines (ex : bourrage papier, erreur de lecture magnétique, coupe inefficace des coupons...).

Service qualification : Ce service faisait des tests d'endurance et d'usure des pièces mécaniques des machines, elle demandait souvent des programmes avec fonctionnalités réduites pour effectuer leurs tests, j'étais amené à proposer des solutions et développer une version spéciale.

Service dessin et reprographie : Le service dessin et reprographie fournissaient à la demande tous les schémas électroniques, les plans de câblage et les plans mécaniques des machines pendant la phase de développement.

Cellule documentation technique : Elle nous fournissait toutes les documentations techniques sur les composants utilisés dans la machine.

4.4.2. Relations externes

Service production à Besançon : Le service production pouvait demander d'intégrer différentes fonctionnalités utiles pour le test de fabrication (ex : tests unitaires des organes électromagnétiques, test magnétique, test d'impression ...) dans ce cas, j'étais amené à proposer des solutions simples qui correspondaient à leur attente, sans mettre en cause l'architecture du programme.

Filiales d'IER à l'étranger : Il n'est pas rare que j'avais eu des contacts avec les commerciaux pour avoir des éclaircissements sur certains points du cahier des charges lors du développement pour un client donné.

5. Projet réalisé

Le projet que je présente concerne le développement logiciel du lecteur spécialisé IER 627BC dédié au contrôle d'accès à bord d'avions, de bateaux et de trains avec reconnaissance d'empreintes digitales.

L'objectif est de faire fonctionner le lecteur spécialisé développé déjà par IER en intégrant deux nouvelles fonctions, la reconnaissance d'empreintes digitales et la lecture du code à barres.

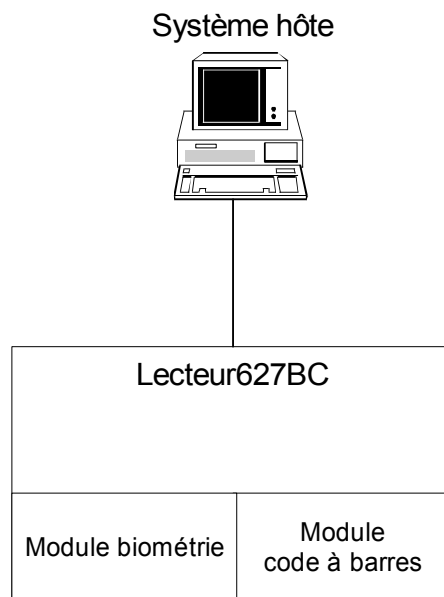
Le projet a officiellement commencé début Mars 2002 et a précédé d'une phase de pré-étude qui a eu lieu au mois de Janvier. Cette pré-étude m'a permis de me former sur le "logiciel de base" du lecteur qui a été une découverte pour moi. Après une phase de spécification et une phase de développement et de test, le projet s'est terminé au mois de septembre 2002.

Mes principales tâches pendant ce projet ont été:

- de définir une solution technique répondant à un cahier des charges et de la réaliser;
- de réaliser le logiciel de la carte interface qui relie le module biométrique et le module code à barres à la carte principale qui contrôle l'ensemble du lecteur IER 627BC;
- de modifier le logiciel de la carte principale pour qu'il prenne en compte ces deux nouvelles fonctions.

5.1. Présentation du lecteur IER 627BC

Le lecteur spécialisé IER 627BC est destiné au contrôle d'accès à bord d'avions, de bateaux et de trains. Il est relié au système hôte via la liaison série RS232, voir la figure suivante.



Le lecteur lit les données sur le billet et remonte les informations concernant le passager et son vol au système hôte et selon la réponse de celui-ci le lecteur doit accepter ou refuser l'accès à bord du passager.

Le lecteur IER 627BC est capable de lire deux types de cartes d'accès à bord, l'une avec pistes magnétiques et l'autre avec code à barres.

Pour contrôler l'accès à bord d'un passager muni d'une carte avec pistes magnétiques, le passager doit introduire sa carte dans la machine, ensuite apposer son index sur le module biométrique intégré. La machine fait alors une comparaison d'empreintes contenues dans les pistes et celles lues par le module. S'il y a concordance et si les informations lues du vol sont correctes alors le passager pourra monter à bord. La figure suivante montre le lecteur 627BC et les différents types de coupons d'accès à bord.

Le passager muni d'une carte avec code à barres doit la présenter devant le rayon laser du module code à barres intégré au lecteur. Le lecteur acceptera le passager si les informations lues sur la carte sont correctes.



Fig. 5 – Lecteur 627BC

5.2. Architecture matérielle du lecteur

Le lecteur est composé d'une carte principale qui possède deux liaisons séries à l'origine. La première liaison est dédiée à la communication avec le système hôte, la deuxième liaison est libre mais elle est insuffisante pour connecter le module biométrique et le module lecteur code à barres, d'où l'adjonction d'une nouvelle carte dite « carte

interface » possédant quatre liaisons séries. Cette carte a pour rôle de transmettre les données d'empreintes digitales et code à barres vers la carte principale ou recevoir les commandes de la carte principale et les transmettre vers les modules.

5.2.1. Carte principale

La carte principale est architecturée autour du microprocesseur type National Semiconductor NS32000 ; une mémoire 512Ko contenant le programme d'application dont j'ai amené à modifier ; une mémoire RAM sauvegardée d'une capacité de 256Ko permettant le stockage des informations de vols ; une mémoire RAM de travail d'une capacité de 256Ko.

Un afficheur rétro-éclairé de 2 lignes de 16 caractères, un clavier et un buzzer permettent l'interaction entre l'utilisateur et la machine.

Un ensemble mécanique équipé deux moteurs pas à pas, une tête magnétique, permet la lecture d'un billet d'accès à bord avec pistes magnétiques.

Deux liaisons séries asynchrones permettent la communication avec le système hôte et la carte d'interface.

5.2.2. Carte d'interface

La carte d'interface est architecturée autour d'un processeur de type INTEL 80186 ; une mémoire de 64Ko contenant l'application dont j'ai eu la charge de développer ; une mémoire RAM de travail d'une capacité de 64Ko.

Quatre liaisons séries asynchrones permettent la communication avec la carte principale, le lecteur biométrique et le lecteur code à barres.

Un module biométrique et un lecteur code à barres permettent respectivement de lire l'empreinte digitale et les codes à barres. Ces données seront transmises à la carte principale pour comparaison via la liaison série dédiée. La figure 6 montre l'architecture matérielle du lecteur 627BC.

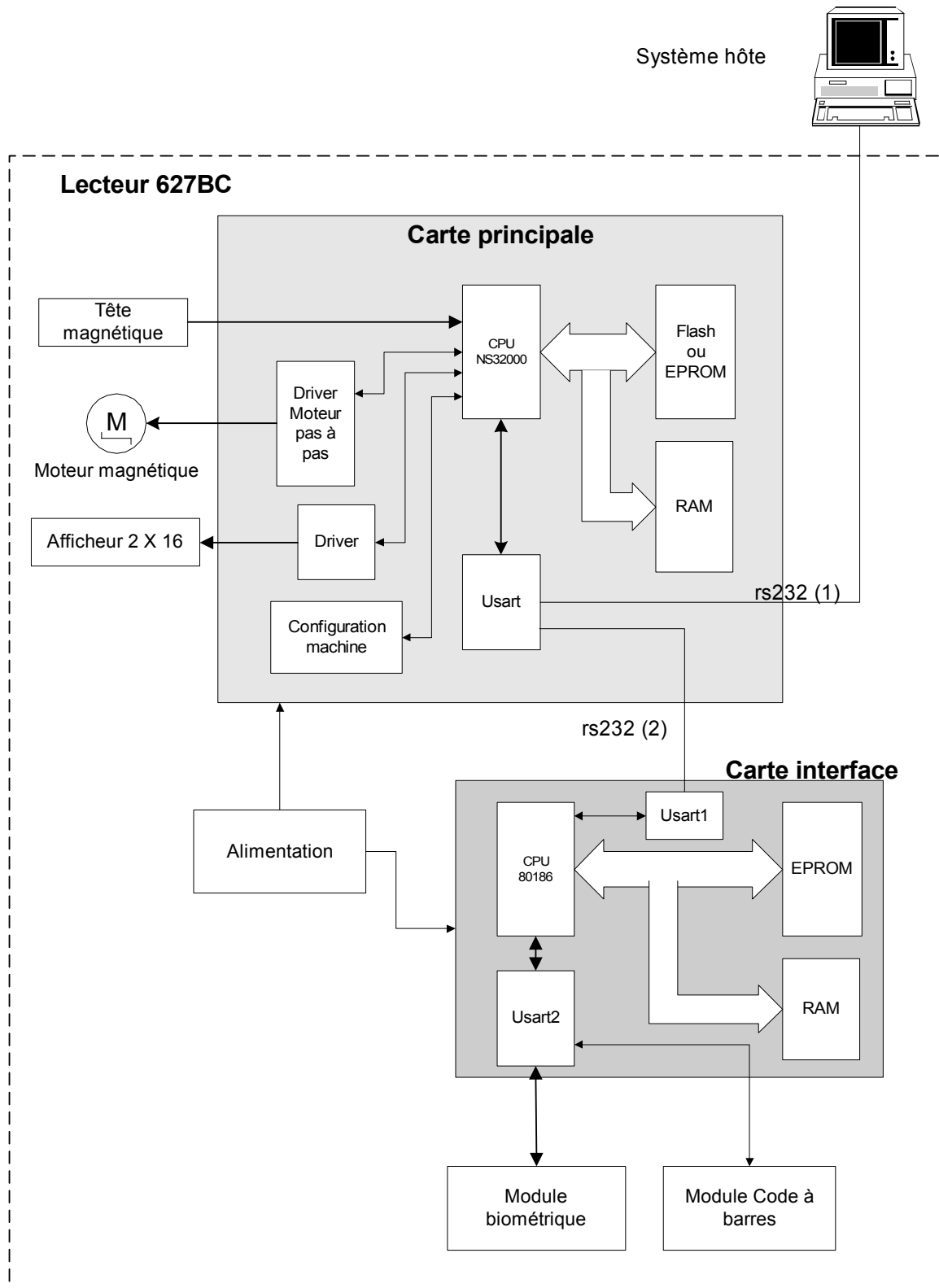


Fig. 6 - Architecture matérielle du lecteur 627BC

5.3. Les contraintes de développement

Les contraintes de développement ont été imposées par le cahier des charges fourni par le chef de projet.

Le cahier des charges a imposé l'utilisation des matériels et logiciels existants afin de minimiser la durée du projet et réduire le coût. Le lecteur IER 627 commercialisé il y a 3 ans a été retenu pour ce projet mais son design a été refait pour intégrer le module biométrique et le module code à barres. Son logiciel qui a été sous-traité en externe à l'origine a été gardé. Je devais le modifier pour qu'il prenne en compte les nouvelles fonctionnalités. Il a été écrit en langage C, utilisant un noyau temps réel. Les ingénieurs qui ont été à l'origine ne sont plus joignables et il y a eu peu de documentations concernant l'architecture du logiciel. Donc, j'ai dû réaliser seule et effectuer cette modification dans un délai très court.

Le cahier des charges a imposé l'utilisation de la carte interface issue d'une ancienne machine dont j'ai développé le logiciel pour prendre en compte la lecture de l'empreinte digitale et la lecture du code à barres.

Il a imposé l'utilisation des chaînes de développements existantes fonctionnant sous UNIX dont les compilateurs C, les éditeurs de liens pour microprocesseurs NS32000 et Intel 80186. L'utilisation des émulateurs existants pour déboguer les logiciels des deux cartes m'a été aussi imposée. La chaîne de développement du microprocesseur NS32000 a été une nouveauté pour moi car il a fallu que je maîtrise son fonctionnement.

Le module biométrique de marque SAGEM et le module code à barres ont été achetés. Je me suis formé sur le fonctionnement de ses appareils ainsi que sur le format leur protocole d'échange afin d'être opérationnelle le plus tôt possible.

5.4. Le cycle de développement

Le cycle de développement présenté a été imposé et contrôlé par le service qualité afin que le projet se déroule conformément aux procédures de développement mises en place dans le cadre de la certification ISO9001 en 2000.

Mon rôle a été de mettre en application ce cycle de développement, de contrôler son déroulement, de valider tous les documents de projet et informer les personnes concernées par l'évolution de ce projet. Ce projet a été suivi parallèlement par mon chef de projet lors de différentes réunions d'avancement.

A partir de contraintes imposées par le cahier des charges, j'ai engagé une pré-étude afin de définir une solution technique. J'ai ensuite rédigé une proposition technique présentant la solution envisagée ainsi qu'une estimation du planning de réalisation. Cette proposition a été étudiée par mon supérieure hiérarchique et mon chef de projet IER 627BC lors de la réunion d'ouverture du projet au cours de laquelle j'ai dû justifier mes estimations dans le but d'obtenir le budget nécessaire au développement du projet.

Après acceptation de la proposition et l'obtention du budget, j'ai élaboré les spécifications fonctionnelles de la carte interface. Ces documents ont eu pour but de décrire les fonctionnalités, les principes de fonctionnement et établir les limites de développement.

Après validation des spécifications, j'ai réalisé la conception détaillée et le codage du logiciel de la carte interface. Une phase de tests unitaires et d'intégration a permis de déverminer le logiciel dans son intégralité. Ensuite, j'ai élaboré une documentation décrivant le fonctionnement minimum du logiciel de la carte principale. J'ai réalisé la conception détaillée concernant le traitement des empreintes digitales et code à barres. Après la phase de codage et de tests unitaires, le lecteur a pu être testé dans son ensemble.

Une phase de recette réalisée en interne par le service de validation a permis de contrôler les fonctions et les performances du lecteur au regard des spécifications fonctionnelles établies en début de projet. Cette phase a permis aussi de déceler quelques anomalies que j'ai corrigées aussitôt.

Le lecteur a finalement été installé chez le client depuis Septembre 2002 et n'a pas eu de remontée de dysfonctionnement de la part du client jusqu'à ce jour.

5.4.1 Avantages et inconvénients du cycle en V

Le processus de développement de ce projet est imposé par le service qualité et suit le modèle de cycle en V (voir figure 7) mais il a un certains avantages et aussi ses inconvénients dans ce modèle que je détaillerai.

En effet, ce modèle tient d'avantage compte de la réalité, le processus de développement n'est pas réduit à un enchaînement de tâches séquentielles. Il montre que:

- c'est en phase de spécification que l'on se préoccupe des procédures de qualification ;
- c'est en phase de conception générale que l'on se préoccupe des procédures d'intégration ;

- c'est en phase de conception détaillée que l'on prépare les tests unitaires ;

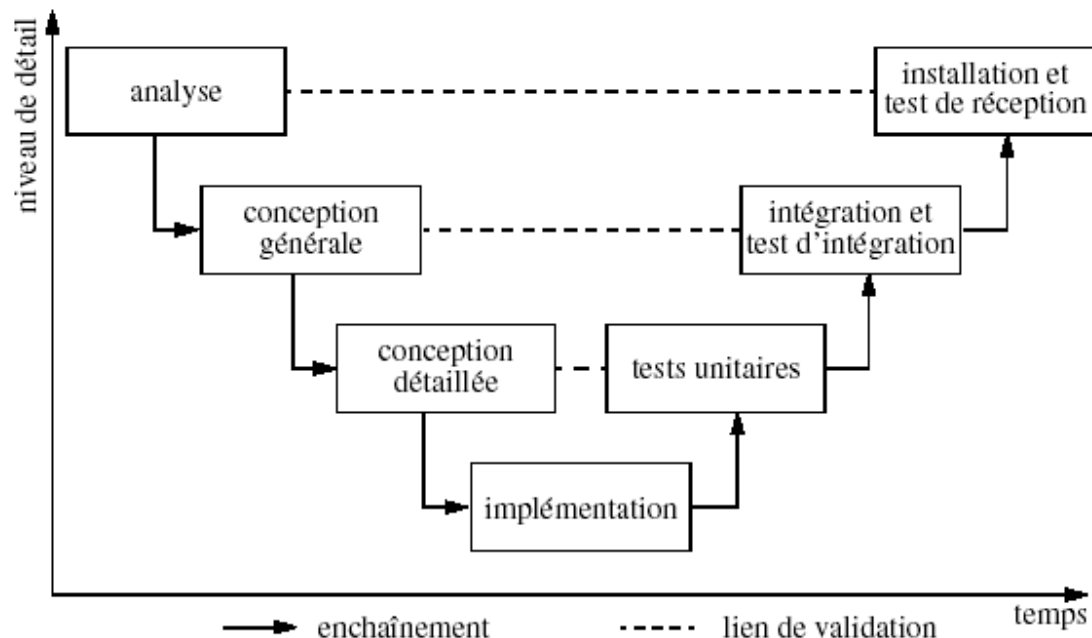


Fig. 7 - Modèle en V du cycle de développement

Le modèle de cycle de vie en V permet d'anticiper sur les phases ultérieures de développement du produit. Il permet de commencer plus tôt le plan de tests de qualification et le plan d'évaluation des performances. Il conduit à commencer plus tôt la documentation utilisateur. Il permet de développer parallèlement différents modules lorsque la phase de conception générale est validée, comme le montre la figure suivante.

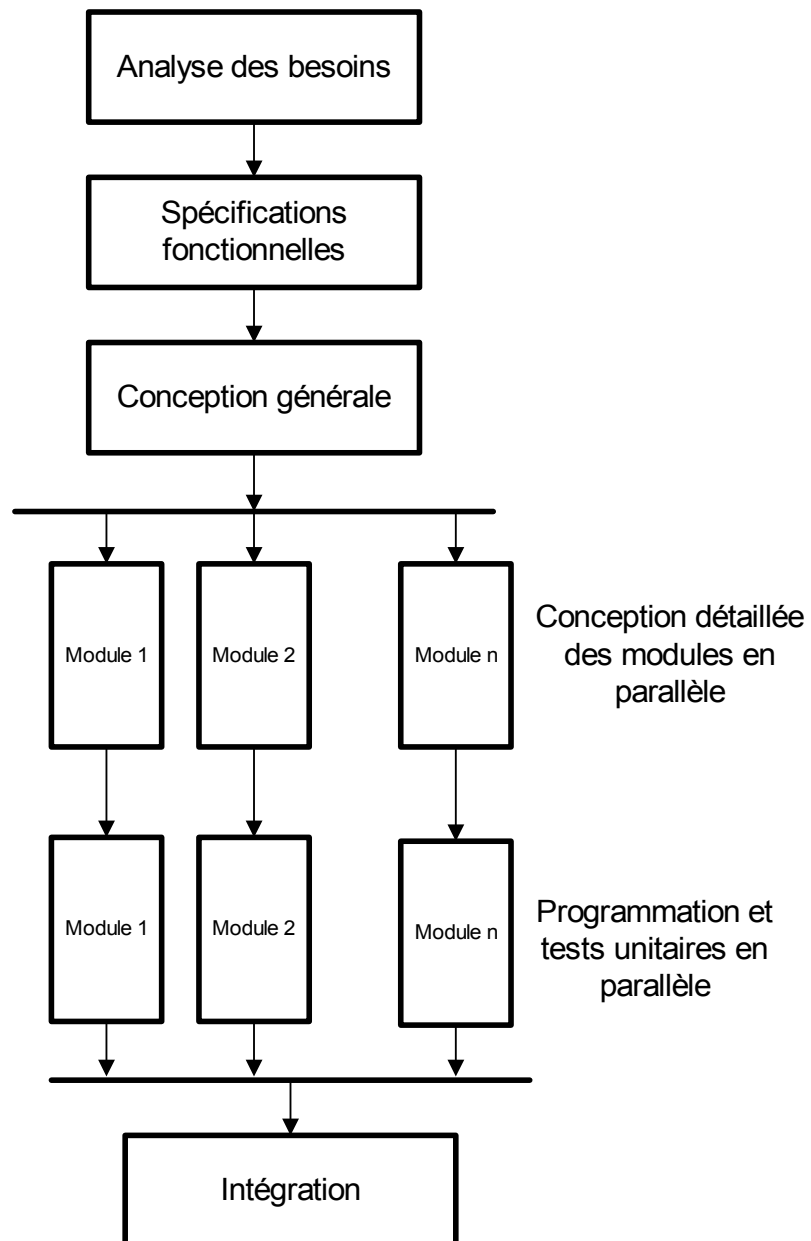


Fig. 8 – Conception détaillée des modules en parallèle.

Ce modèle de développement permet de contrôler les rétro-actions ; la vérification/validation par une critique constructive évite les retours arrière ; le cycle de vie met l'accent sur les phases amont par rapport à la programmation, c'est à dire la spécification et la conception.

Toutefois, le modèle a ses limites et est parfois difficile à appliquer rigoureusement. En effet, la preuve effective du bon fonctionnement du produit n'est réalisée que tardivement dans le cycle – lors de la phase d'intégration – lorsqu'il est possible d'évaluer concrètement le logiciel. Avant cette phase d'intégration, seuls les documents ont été produits. Or, ce n'est pas parce qu'un document passe avec succès sur une étape de validation sur papier que le logiciel qu'il décrit donnera forcément des résultats convaincants.

En fait, la démarche du cycle en V ne donne des résultats satisfaisants que lorsqu'il est effectivement possible d'enchaîner les phases sans trop de problèmes. Il faut que l'ensemble des besoins soit parfaitement connu et le problème complètement compris par les analystes ; il faut aussi que la solution soit facile à déterminer par les concepteurs et le codage se réduit à la génération automatique de code, à partir des documents de conception.

Hélas, les projets ne se présentent pas tous sous ces conditions idéales. Force est de constater que la part inconnu, et donc de risques, peut être grande dans certains développements, notamment du fait :

- de la méconnaissance des besoins par le clients,
- de l'incompréhension des besoins par le fournisseurs,
- de l'instabilité des besoins, qui découle des deux points précédents,
- des choix technologiques...

Pour toutes ces raisons, des retours d'information entre les phases sont nécessaires pour incorporer des corrections en amont, en fonction de découvertes réalisées en aval. Ces retours entre les phases perturbent la vision linéaire donné par le cycle de vie en V.

En fait, les retours entre les phases ne sont que le reflet de la réalité. Tant que ces retours restent limités entre les phases adjacents, le modèle en V conserve toute sa pertinence. Dans le cas contraire, le modèle n'est plus vraiment adapté. Il faut en changer et non essayer de faire coller la réalité au modèle.

Malgré ces inconvénients, le modèle est quand même retenu car les besoins clients sont clairement identifiés, le projet n'est pas très complexe et d'ailleurs il a été imposé par le service qualité.

5.5. Processus de développement

J'ai donc été chargé de m'occuper du développement en langage C de la carte interface du lecteur dont son rôle est de transmettre les données lues d'empreinte digitale ou du code à barres provenant respectivement du module biométrique ou du module code à barres, vers la carte principale. La carte d'interface est capable de recevoir les commandes provenant de la carte principale est de les transmettre aux modules.

5.5.1. Définition du protocole d'échange de la carte interface

Avant de me lancer dans le développement logiciel de la carte interface, j'ai d'abord élaboré un document décrivant le protocole d'échange entre la carte interface et la carte principale. Le protocole que j'ai mis en place est un protocole avec checksum et acquittement, il est assez simple à implémenter et efficace, c'est le protocole le plus utilisé dans l'industrie.

Ensuite, j'ai élaboré un document décrivant les messages qui transitent entre la carte interface et la carte principale. Etant donné que les messages peuvent transiter dans les deux sens, c'est à dire de la carte principale vers les deux modules via la carte interface, ou des modules vers la carte principale via la carte interface, il est évident qu'il y a une sorte d'identifiant dans les messages permettant la reconnaissance aisée de l'expéditeur ou du destinataire.

Ces deux documents ont été indispensables pour comprendre les échanges des données entre les cartes et pour développer les drivers. Je les ai diffusés aux personnes concernées, à mon chef de projet et à mon supérieur hiérarchique et ils ont été approuvés par le service qualité.

5.5.2. Développement logiciel de la carte interface

Pendant la phase de développement j'ai été le seul intervenant pour la conception détaillée, le codage et le test d'intégration du logiciel.

5.5.2.1. Conception détaillée

Lors de la conception, j'ai défini l'architecture du logiciel de la carte de façon à obtenir une structure évolutive et réutilisable qui faciliterait la tâche d'un tiers pour faire évoluer le logiciel ou qui réduirait le temps de développement des futures applications.

Pour que cette structure soit évolutive, je l'ai découpée en plusieurs processus de tel manière à les rendre le plus indépendants possibles l'un par rapport aux autres et lorsqu'il y avait une modification dans un processus, il ne fallait pas qu'il y ait une interférence néfaste ou un dysfonctionnement sur les autres.

Etant donné que le but de la carte interface était d'acheminer les messages des modules vers la carte principale du lecteur et vice versa, il m'a paru important de bien séparer l'ensemble des processus d'acheminement d'un module par rapport à l'autre. J'ai pris un soin bien particulier à ce que ces deux ensembles de ces processus soit bien indépendants les uns des autres.

Ensuite, j'ai découpé les processus précédemment identifiés en plusieurs fonctions. Le choix des noms que j'ai adoptés pour nommer ces fonctions et les variables sont assez implicites, ils rappellent automatiquement le processus incriminé et font un lien direct dans le document de conception que j'ai élaboré. Cette technique permettrait à un tiers de rentrer dans le code du programme facilement, d'avoir une vue globale du logiciel et d'être opérationnel rapidement.

Par expérience, lorsqu'un développeur n'a pas une vue globale sur le logiciel auquel il a amené à modifier, il a toujours tendance à créer de nouvelles variables ou de nouvelles fonctions pour corriger l'anomalie. Ce genre de procédé permet de corriger ponctuellement l'anomalie mais il crée surtout d'autres anomalies qui se manifesteront plus tard et même beaucoup plus tard, surtout chez le client. Dans le jargon informatique, on appelle cela « des verrues ». C'est pour cela que les noms des fonctions et des variables et les commentaires sont très importants ainsi que les documentations.

5.5.2.2. Codage

Lorsque l'analyse et la conception ont été réalisées de façon claire et explicite, le codage est devenu alors aisé et sans encombre. Concernant le format des fichiers, j'ai appliqué les règles en rigueur du service. J'ai pris soin de bien commenter les entêtes de fichiers, les entêtes des fonctions et des variables. Toutes les fonctions que j'ai créées ne pas dépassent pas une page d'écran en terme de lignes. Par expérience, ce genre de découpe permet de garder le fil conducteur de compréhension car on voit toutes les variables locales et globales utilisées par la fonction aussi on voit sa structure complètement, sur une seule page alors que si une fonction tient sur dix pages d'écran voire plus, il est impossible de comprendre toute la structure d'un seul coup d'œil ; on se pose toujours des questions sur l'utilité des variables ; il faut jongler entre les pages d'ou perte de temps et de fil conducteur.

J'ai commencé par coder les fonctions de bas niveau afin de faire abstraction des différents registres, des entrées sorties, et adresses liées au matériel que j'ai nommé la couche physique. Ensuite, j'ai construit petit à petit la couche de liaison en me basant sur ces différentes fonctions. Cette couche m'a permis par exemple de transmettre ou recevoir une trame d'un des modules sans me soucier des contraintes liées aux matériels. Une fois que cette couche a été codée, j'ai commencé à écrire les drivers des modules. Ces drivers sont capables d'émettre ou de recevoir des données avec la gestion de protocole conformément au document que j'ai établi. Enfin, j'ai fini le codage par la couche application qui est composée de différents processus que j'ai décrits dans la phase de conception. Le codage du logiciel de la carte interface m'a pris 4 semaines.

Tous les fichiers que j'ai créés pour le logiciel de la carte interface sont stockés dans un répertoire du serveur et sont classés par projet. En dessous de ce projet, j'ai les organisé en sous-répertoires en adoptant une logique de processus c'est à dire que je les ai nommés selon le nom donné aux processus décrits dans le document de conception. Ce type d'organisation de fichiers permet aux nouveaux arrivants dans le projet de localiser facilement l'emplacement des fichiers et aussi à moi même lorsque j'aurais amené à le modifier plus tard.

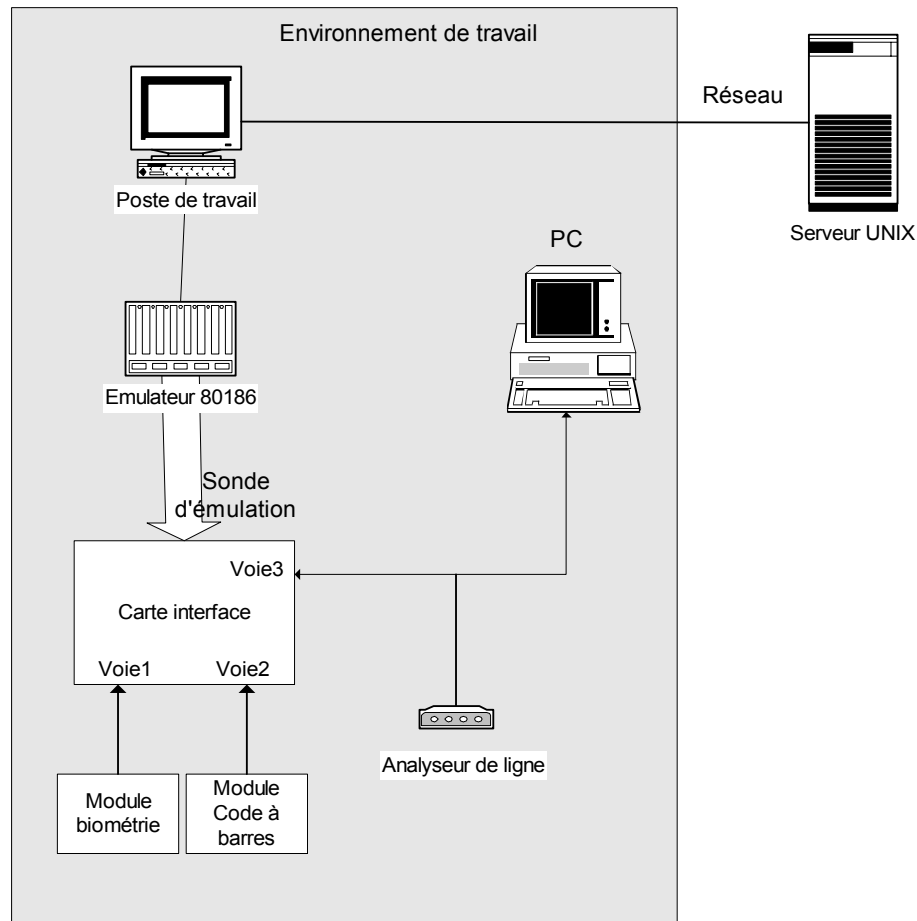
5.5.2.3. Tests de la carte interface

Quelques jours avant que le codage n'arrive à terme, j'ai signalé à mon chef de projet que j'aurais besoin :

- d'un émulateur,
- d'une station de travail,
- d'un PC,
- d'un analyseur de ligne,
- de la carte interface,
- du module biométrique,
- du module code à barres,

pour commencer à tester mon application pour une durée déterminée comme définie dans le planning que j'ai établi. Etant donné que le nombre d'appareils était insuffisant pour tout le service logiciel, j'ai été obligé de les réserver à l'avance. Le poste de développement a été déployé comme indiqué sur la figure ci-dessous.

Fig. 8 - Environnement de développement



Le PC m'a servi à simuler l'émission, réception des données des modules biométrique ou/et code à barres et l'analyseur de ligne m'a permis de visualiser des données transitant sur les liaisons séries. L'émulateur a servi de substitution au microprocesseur, il permet de visualiser les mémoires, les registres, de visualiser les instructions exécutées du programme etc... Et le poste de travail m'a servi à l'édition, à la compilation du logiciel et le débogage de mon application.

5.5.2.4. Tests unitaires et intégrations

Le test de mon logiciel a débuté d'abord par toutes les fonctions de bas niveau, j'ai vérifié que les circuits périphériques de la carte étaient bien configurés et que l'émission réception sur chaque canal fonctionnait correctement et que les fonctions annexes

faisaient bien leur rôle. Cette vérification a été réalisée par visualisation sur l'analyseur de ligne et par le PC de simulation. Ces tests unitaires sont très importants car ils m'ont permis de valider la couche physique ainsi que le hardware de la carte et par la suite ils m'ont permis de « ne plus mettre en cause » cette couche lors du débogage des couches supérieures. Ensuite, j'ai attaqué les tests de la couche dite de liaison ; c'est une couche un peu plus évoluée par rapport à la précédente. J'ai vérifié que la carte pouvait recevoir et transmettre une trame via les différentes liaisons quelles que soit les vitesses de transmission (ex : 9600 bauds à 56 Kbauds), ce qui a prouvé que la mise en place des interruptions de caractères que j'avais adoptée a été efficace. J'ai mesuré à l'oscilloscope la durée de traitement de ces interruptions et elle était très faible.

Par la suite, j'ai testé les drivers envoyant par le PC les messages dits « normaux » puis des messages avec défauts (erreur de parité, erreur de checksum...). Le logiciel a réagi conformément au document de tests unitaires. J'ai aussi transmis des messages de grandes tailles à une vitesse de 56 Kbauds. Le driver a réagi efficacement, sans perte de caractères, ce qui prouve que les fifos (first in first out) de réception sont bien dimensionnés. Un fifo de réception est une sorte de buffer tampon qui permet de stocker des caractères reçus à un rythme très élevé généralement sous interruption en attendant que le programme principale (ou la tâche de fond) viennent le vider selon sa cadence. Le test du protocole à fréquence élevé m'a permis de vérifier la limite de mon logiciel, en pratique le protocole utilisé est de 9600 bauds fixé par les deux modules.

Pour tester la couche application, j'ai branché les deux modules sur les deux premières liaisons séries et un PC de simulation sur la troisième liaison série ; ensuite j'ai provoqué la lecture de l'empreinte digitale puis la lecture du code à barres et j'ai vérifié les données transmises vers le PC. En provoquant un envoi simultané de l'empreinte digitale et du code à barres, j'ai vérifié que les données étaient bien transmises l'une après l'autre vers le PC. Cela a prouvé que la gestion de priorité de messages que j'avais mis en place avait bien fonctionné. Les tests d'intégrations n'ont pas été réalisés car le logiciel de la carte principale du lecteur n'était pas fonctionnel mais ils ont été prévus plus loin dans le planning.

Grâce à la séparation de mon logiciel en couches (physique, liaison, etc..), l'ensemble des tests ont pu se dérouler sans problème majeur.

5.5.3. Développement logiciel de la carte principale

Après les tests unitaires et d'intégration de la carte interface, j'ai commencé à analyser le logiciel de la carte principale que je ne connaissais pas car il a été développé par une société de sous-traitance il y a 3 ans. Il y avait peu de documentations concernant l'architecture du logiciel et il y avait personne en interne qui était capable de me

renseigner sur son fonctionnement. Donc, j'ai perdu beaucoup de temps à analyser et à faire des documentations pour mieux comprendre l'architecture.

La durée de développement de cette partie que j'avais prévue était sur le point de déborder. J'ai alors signalé mon chef de projet la nécessité de prendre une ressource supplémentaire afin de pouvoir tenir le délai de livraison du lecteur. Je lui ai laissé entendre que les heures supplémentaires et les week-ends seraient peut être nécessaires dans les jours ultérieurs. Malgré quelques réticences, ma demande a finalement été acceptée et un ingénieur de développement a intégré l'équipe, il a eu la charge de concevoir le driver du lecteur. J'ai alors rectifié mon planning en conséquence.

Avec l'ingénieur, nous avons adopté une stratégie pour accélérer l'analyse et la conception ; nous avons recensé tous les modules concernés par les modifications. Ensuite, nous avons pris en charge chacun la moitié de la liste et nous avons réalisé une documentation décrivant le fonctionnement de la partie réception, émission, la partie traitement de billet magnétique. Lorsque nous avons fini l'analyse et la documentation, nous avons codé chacun sa partie.

5.5.3.1. Test unitaires et d'intégrations du lecteur

Lorsque le codage a été terminé, j'ai réalisé des tests unitaires et d'intégrations du lecteur.

Pour faire les tests d'intégrations, j'ai eu besoin d'un PC simulant le système hôte utilisant un logiciel appelé en interne le « simulateur host ». Ce logiciel est capable de créer et d'exécuter des macros, ce qui rend les tests automatiques. J'ai créé des macros pour les tests d'intégrations du lecteur, ces macros sont toujours accompagnées de fiches de tests et sont sauvegardées sur le réseau intranet afin que les personnes concernées du projet puisse accéder et l'utiliser.

Les tests d'intégrations m'ont permis de corriger quelques bogues mineurs du lecteur. Lorsque le lecteur a été validé au niveau du service logiciel, il était ensuite validé par la cellule de validation qui est le représentant de l'utilisateur final. Cette cellule valide toutes les machines sortant du service, au point de vue mécanique, électronique et logiciel, avant la livraison aux clients.

5.6. Degré d'avancement

Le planning a subi plusieurs des modifications au cours du projet. Ces modifications ont été dues essentiellement à des retards importants dans la phase d'analyse et de

formation du logiciel de la carte principale. Grâce à ma demande d'une ressource supplémentaire et aux efforts que moi même et l'ingénieur de développement avons fournis tout au long du projet, le lecteur a pu livrer à temps. Nous avons fait des heures supplémentaires et travaillés les week-ends durant deux mois pour terminer du logiciel.

Le lecteur a finalement été installé sur site réel en Septembre 2002 à la date prévue. Il satisfait toutes les contraintes imposées par le cahier des charges.

5.7. Bilan technique

L'utilisation du langage C dans ce projet m'a permis d'approfondir mes connaissances en programmation structurée, de mieux appréhender l'utilisation des structures, des pointeurs et des pointeurs de fonctions. L'encapsulation ou l'assemblage de toutes les fonctions qui manipulent la même structure de données dans un même fichier (modules) facilite la lecture du programme dont améliore la compréhension.

Parallèlement, durant toute la phase d'analyse et de conception du logiciel, je me suis formé sur le fonctionnement du microprocesseur NS32000 de National Semiconductor, sur l'utilisation de l'émulateur NS32000 et sur l'utilisation de la chaîne de développement (compilateur, assembleur, linker...) afin d'acquérir toutes les connaissances pour être opérationnel lors du codage et des tests du lecteur.

L'utilisation du noyau temps réel dans le lecteur m'a permis de prendre conscience de son importance. En effet, les systèmes temps réel sont, dans la plupart des cas, des systèmes intégrant le noyau temps réel. Ces systèmes possèdent des contraintes temporelles fortes. Lors de l'arrivée d'un stimulus (événement) en provenance de l'environnement externe, le système doit réagir dans un délai donné. Il permet aussi de séparer le logiciel en tâches et de les distribuer à l'équipe de programmeurs pour qu'ils puissent les concevoir et les coder indépendamment.

J'ai analysé le logiciel du lecteur et de la carte interface en utilisant la méthode SA-RT qui est une technique de description fonctionnelle permettant de spécifier un système embarqué temps réel tant du point de vue logiciel que matériel (architecture). SA-RT définit des principes de mise en place d'un modèle d'exécution multi-tâches temps réel et propose des principes intéressants que je détaillerai dans le chapitre optimisation mise en oeuvre.

Lors de l'implémentation, le logiciel est écrit d'une part dans un langage de programmation comme le langage C et d'autre part s'appuie sur les services d'un noyau temps réel qui fournit au concepteur tous les outils (primitives) pour contrôler

l'environnement logiciel des tâches. L'utilisation de la programmation multitâches permet d'introduire la notion de priorité d'exécution qui, couplée à la préemption, permet d'affiner les performances temporelles (temps de réponse) de l'application. L'utilisation des techniques basées sur les algorithmes d'ordonnancement temps réel permet en outre de valider l'application d'un point de vue temps réel (respect des délais).

Le logiciel sous-traité de la carte principale m'a permis de comprendre combien il est important de ne pas écrire des fichiers trop longues (>2000 lignes) et des fonctions de plus de dix pages d'écran, d'éviter d'avoir énormément de variables globales utilisées dans tous les sens, d'avoir des documentations.

5.8. Optimisation mise en oeuvre

5.8.1. Optimisation documentaire

Les documents du projet ont permis d'établir des limites du projet et les méthodes de validation afin que le lecteur réponde aux besoins du clients. Les principaux documents de ce projet sont :

- les spécifications fonctionnelles dont l'objectif étaient de poser clairement les limites du développement en accord avec les besoins identifiés ;
- les documents de conception détaillée dont le but étaient de décrire les différents processus du logiciel ;
- les fiches de test permettant de valider les tests d'intégrations du lecteur ;
- le cahier de recettes renseigné qui, en consignait les résultats de la qualification, a permis de certifier que le lecteur avait été contrôlé et satisfait aux spécifications.

J'ai fait en sorte que les documents soient compréhensibles par toute personne interne ou externe au service. J'ai pré-formaté cette documentation suivant le système de qualité de sorte qu'elle ait une structure facilitant la lecture et la compréhension.

Ces documents ont été approuvés par le service qualité et par moi-même. Certains de ces documents sont contractuels et ont été aussi approuvés par mon supérieur hiérarchique ainsi que par le chef de projet lecteur IER 627BC.

5.8.2. Optimisation au niveau du développement

5.8.2.1. Optimisation en couches

Ce projet m'a permis de perfectionner mes compétences en analyse et conception de logiciel. Le fait de concevoir un logiciel en différentes couches permet de fractionner la complexité du problème en couches plus élémentaires donc de faciliter la compréhension et le déverminage du logiciel.

5.8.2.2 Optimisation par la méthode SA-RT

La méthode SA-RT fut développée par D.J. Hatley and I.A. Pirbhai afin de spécifier et de concevoir des systèmes temps réel. Elle répond aux besoins de ces systèmes en intégrant l'analyse structurée de DeMarco et la théorie des machines à états finis pour la représentation d'une structure des processus et d'une structure de contrôle.

La modélisation en SA-RT se base sur la décomposition du système en processus. Un processus est une entité dont le but est simplement de produire ses sorties à partir de ses entrées. Ces entrées et ces sorties constituent ce que l'on appelle les données. Lorsque le système exécute un processus, on dit qu'il est dans un état. Le passage d'un processus à un autre (ou l'activation d'un processus) se fait par la vérification de certaines données appelées données de contrôle. Ces données sont spécifiées à l'aide de machines à états et de tableaux d'états.

La méthode SA-RT s'avère efficace pour des programmes moyens de 5000 à 50000 lignes ce qui le cas de ce projet.

6. Composantes du métier d'Ingénieur-Maître

Le poste de responsable logiciel du lecteur a impliqué d'assurer plusieurs fonctions pendant le développement. Cette activité m'a permis d'assurer à la fois :

- des fonctions techniques consistant à assurer le développement logiciel du lecteur ;
- la fonction de gestion de projet et d'encadrement ;
- la fonction de transmission ;
- la fonction financière.

6.1. Fonctions techniques

Mon professionnalisme et mon sérieux dans le travail ont toujours été reconnu par mon supérieur et c'est la raison pour laquelle mon chef de service m'a choisi comme responsable développement logiciel de ce projet.

Ma formation pluridisciplinaire et mon expérience professionnelle m'ont permis d'acquérir les compétences complémentaires dans le domaine informatique, électronique et mécanique pour évaluer les performances d'un logiciel, la faisabilité d'un projet, la fiabilité d'un produit ou d'un système mécanique, pour évaluer les coûts et les délais de réalisation d'un projet et pour apporter des solutions techniques satisfaisants aux contraintes d'un cahier des charges.

Ma formation en informatique et mon expérience en développement logiciel acquis sur les différents machines d'IER m'ont permis de développer les compétences nécessaires pour concevoir des logiciels structurés et fiables. L'utilisation de la méthode SA-RT pour concevoir le logiciel m'a permis de maîtriser la complexité du programme mais surtout elle va permettre aux futurs intervenants de se former rapidement au logiciel.

Ma formation en électronique et système numérique m'a procuré la connaissance nécessaire pour comprendre le fonctionnement de l'électronique du lecteur à partir de schémas qui sont la base de travail pour le développement logiciel. Elle m'a permis aussi de dialoguer en terme métier avec les ingénieurs en électronique et aussi d'analyser les défauts de fonctionnement de la carte.

6.2 Fonctions de gestion et d'encadrement

Etant le plus expérimenté sur l'imprimante durant mon activité au sein du service, j'ai été nommé responsable du « logiciel de base » afin d'améliorer les performances, effectuer le suivi des anomalies et ajouter de nouvelles fonctionnalités selon les demandes clients. Ce logiciel a servi ensuite comme point de départ pour développer toutes les applications spécifiques à différentes compagnies aériennes et ferroviaires. J'ai eu donc sous ma responsabilité plusieurs ingénieurs de développement. Je leur ai apporté des conseils d'expertises concernant leur projet. J'ai eu à estimer les délais de leur développement selon le cahier des charges et établir avec mon chef de service leur planning. Ensuite, à chaque fin de semaine, j'ai réuni tous les développeurs pour discuter de l'état d'avancement de leur développement afin de rendre compte ensuite à mon chef.

6.3. Fonction de transmission

En tant que responsable du « logiciel de base » de l'imprimante à travers mes activités, j'ai une vue globale du logiciel et j'ai une parfaite connaissance sur toutes les applications spécifiques. C'est la raison pour laquelle mon chef de service m'a choisi pour former les nouveaux ingénieurs intégrant dans l'équipe. Je leur ai décrit l'architecture du logiciel, présenté les stations de travail, le serveur et le poste de travail, décrit l'organisation des fichiers constituant le logiciel, formé sur le fonctionnement de l'imprimante.

6.4. Fonction financière

Durant le projet du lecteur, j'ai eu à négocier des moyens humains auprès de mon hiérarchie. En effet, lorsque j'ai senti que la tâche « développement carte principale » prévue dans le planning allait déborder et qui aura pour conséquence le décalage de la date de livraison, j'ai alors signalé mon chef de la nécessité d'une ressource supplémentaire pour me seconder dans le développement du driver de la carte principale.

7. Thème d'étude : La problématique du logiciel

7.1. Le génie logiciel

La production de logiciel tout comme celle de n'importe quel autre produit nécessite la mise en œuvre de méthodes, techniques et outils dépassant largement le cadre de la seule programmation, regroupés sous le nom de Génie Logiciel.

Le terme a commencé à se répandre dans les années 80, notamment avec l'arrivée sur le marché d'ateliers de génie logiciel (AGL). Le Génie Logiciel est à rapprocher du Génie civil ou Génie mécanique. Par exemple, la réalisation d'un pont ne peut être menée sans méthodologie, de même la réalisation d'un logiciel nécessite un minimum de précautions qui garantissent sa faisabilité, sa qualité et sa fiabilité.

La programmation est une discipline récente purement intellectuelle, on y trouve des attitudes et des habitudes qui paraîtraient étranges par rapport à d'autres disciplines plus mures. On n'imagine pas un ingénieur mécanicien réinventer la roue ou fabriquer ses propres vis et boulons alors qu'un programmeur préfère souvent réécrire ses propres lignes de code plutôt que réutiliser un code développé par d'autres.

La lutte contre cette tendance naturelle est un défi majeur de l'industrie informatique d'aujourd'hui connu sous le terme réutilisation.

La programmation est une activité immatérielle : pour un néophyte, il peut sembler peu dispendieux de démolir et reconstruire une architecture logicielle alors que l'on réfléchirait à deux fois avant d'agir de même concernant une autoroute ou un bâtiment.

En conclusion le logiciel est aujourd'hui présent partout, sa taille et sa complexité augmentent de façon exponentielle, les exigences de qualité sont de plus en plus sévères, la crise du logiciel apparue dans les années 70 n'est toujours pas franchement résolue même si de grands progrès ont été réalisés dans de nombreux domaines.

7.2. La crise du logiciel

La crise se caractérise par l'absence de maîtrise des projets, au niveau des coûts et des délais, la mauvaise qualité des produits: les produits ne répondent pas aux besoins définis et des erreurs résiduelles persistent dans le produit final un stock important de projets en attente faute d'une gestion rigoureuse.

Quelques exemples de logiciels défaillants:

- La facture de 0F (juste pour rire)
- La fausse attaque de missiles (Novembre 1979)
- Les anti-missiles Patriotes (Guerre du Golfe 1991)
- Les sondes perdues vers Vénus (années 60) et vers Mars (1999)
- L'échec d'Ariane 5 (1996)
- Et l'an 2000 (qui a fait dépenser des millions de \$ pour des logiciels développés pour la plupart en pleine crise du logiciel) qui a finalement été bien absorbée sans provoquer trop de défaillances.

IER a eu aussi une crise logicielle dans les années 88 à 90. En effet, à l'époque il n'y avait pas de cellule de validation, les machines sont livrées aux clients sans être testées de façon rigoureuse ce qui se traduisait par des mises à jour logicielles sur site qui sont très coûteuses pour la société et à terme elle risquerait de perdre sa clientèle.

Je peux citer un exemple de défaillance survenu aux lecteurs installés à Roissy en 1992 qui était une année bissextile. Le 29 Février, tous ces lecteurs refusaient les voyageurs à l'embarquement car ils considéraient que c'était le 1^{er} Mars. Cet incident a donc provoqué un mécontentement de la part des hôtesses de l'air. Au lendemain, il y a eu une réunion d'urgence des services concernés pour discuter sur les mesures à adopter afin que ce genre de défaillance ne se reproduise plus.

7.3. Les réponses à la crise

7.3.1 La maîtrise du processus de développement

Pour maîtriser le développement, une entreprise doit choisir et appliquer un modèle de processus définissant:

- Les phases du développement: définitions des besoins, spécifications, planification, conception, codage, tests ;
- Les critères de changement de phase ;
- Un cadre pour la gestion de projet.

L'accent est mis sur les phases amont : analyse, conception et non plus seulement sur la programmation.

La société IER a su mettre en place ces processus à partir de 1993, pour améliorer la qualité de ses logiciels. Mon rôle, pour ce projet, a été de mettre en application ces processus de développement, de contrôler son déroulement afin d'atteindre l'objectif fixé c'est à dire respecter les coûts et les délais de développement et d'améliorer la qualité du logiciel.

7.3.2. Développement de méthodes structurées

Dans les années 75-90 les méthodes structurées se développent (SA/RT, SADT...) et beaucoup utilisées dans les systèmes embarqués. J'ai mis en application la méthode SA/RT dans ce projet car j'ai eu une formation sur cette méthode et aussi elle s'applique parfaitement au système temps réel et au programme de taille moyenne, aussi la méthode est nécessaire pour:

- maîtriser la complexité du problème informationnel à résoudre;
- sortir la construction des systèmes d'information de l'empirisme individuel et la fonder sur une coopération efficace entre informaticiens et gestionnaires;
- permettre la communication entre individus de l'équipe de conception;
- construire des systèmes pertinents, fiables, flexibles et adaptatifs;
- permettre d'évaluer le système à tout moment de son cycle, tant sur le plan de son efficacité technique que sur celui de sa pertinence par rapport aux besoins des gestionnaires;
- améliorer les coûts, les délais et la productivité des activités de développement.

7.3.3. Le paradigme objet

Dans les années 90 la crise n'est toujours pas résolue. Si les méthodes structurées s'avéraient efficaces pour traiter des programmes de 5000 à 50000 lignes, elles ne le sont plus pour des produits de 500.000 voire 5.000.000 lignes.

Or les développements sont de plus en plus complexes, de plus en plus longs et sont obsolètes de plus en plus rapidement. Il y a donc une nécessité à réutiliser les connaissances et à les partager.

Dans le domaine de la maintenance également, les méthodes structurées n'ont pas fait leurs preuves. Le problème vient du fait que les méthodes structurées sont soit orientées données soit orientées actions et que la modification d'un élément du logiciel impacte d'autres éléments de manière imprévisible.

Un nouveau paradigme voit le jour en 80 et devient très populaire dans les années 90, c'est le paradigme objets qui encapsule actions et données. Ceci facilite la maintenance car la localisation de l'erreur se fait plus facilement. L'objet masque la structure interne des données qui sont accédées uniquement à travers des méthodes. Chaque objet est autonome. En outre la définition de l'héritage et de la généricité permet la réutilisation d'objets déjà développés dans un autre contexte.

Pour les nouveaux développement, la société IER a mis en place une nouvelle politique qui consiste à analyser en langage UML, à concevoir en C++ et utiliser des outils de génie logiciel. J'ai suivi une formation en UML et en C++ pour développer différentes parties du programme de l'imprimante nouvelle génération et j'ai été responsable logiciel de cette imprimante avant d'intégrer le projet du lecteur.

7.3.4. L'approche composants

Les années 2000 seront placées sous le signe de la programmation à base de composants réutilisables. Cette approche peut être vue comme une forme ultime de la programmation par objets, ou plus simplement comme un assemblage de briques logicielles prédéfinies, orientés techniques ou métiers, et conçues dans le but d'être réutilisées (ce qui n'est pas forcément le cas d'objets définis dans un contexte particulier et réutilisés par hasard).

Cette démarche s'appuie sur les technologies COM, active X etc., mais nécessite une réorganisation du processus de développement logiciel.

En conclusion, on peut dire que si la crise du logiciel est aujourd'hui résorbée parce que mieux connue, elle est loin d'être terminée. C'est que les avancées rapides de la technologie rendent possibles des applications jusqu'alors inenvisageables et obligent sans cesse à reconsidérer les vérités que l'ont croyait acquises.

L'informatique est aujourd'hui une industrie mure pour laquelle des éléments d'assurance qualité peuvent être mis en place et qui diffèrent selon le type de produit à fabriquer et le contexte dans lequel le produit sera utilisé : les contraintes seront différentes selon que l'on fabrique un logiciel amateur ou un logiciel professionnel.

8. Conclusion technique

Le lecteur 627BC a été réalisée afin de répondre à la demande client suite à l'événement du 11 Septembre 2001. J'ai été choisi pour être responsable de développement de ce lecteur. En qualité de chef de projet, j'ai été responsable de la gestion du projet, de l'encadrement encadré d'un ingénieur et de l'application des processus de développement imposé par le service de qualité dont les principales phases ont été la phase pré-étude, la phase développement et la phase test de validation.

L'analyse du programme de la carte interface effectuer avec la méthode SA-RT, m'a permis de maîtriser la complexité du problème à résoudre ; de construire un système pertinent, fiable, flexible et adaptatif ; d'améliorer les coûts, les délais et la productivité des activités de développement ; de permettre la communication entre individus de l'équipe de conception. Le développement logiciel du lecteur en langage évolué et le découpage du logiciel en différentes couches ont rendu le logiciel plus robuste, plus fiable et portable.

En qualité de responsable logiciel du lecteur, mes compétences en informatique, électronique et mécanique m'ont permis de trouver des solutions techniques satisfaisant aux contraintes imposées par le cahier des charges. Je suis intervenu pendant le développement pour réaliser la conception du logiciel de la carte interface et la modification du logiciel du lecteur.

9. Conclusion générale

Mon activité professionnelle en tant que responsable du développement logiciel du lecteur au sein du service logiciel, m'a permis à la fois de mettre en oeuvre mes compétences techniques et d'élargir mon champ de compétences en gestion de projet.

En tant que responsable du logiciel, j'ai assisté et participé à différentes réunions d'avancement et j'ai pu donner les solutions sur divers problèmes techniques rencontrés durant le projet aussi j'ai donné des idées sur les différentes applications spécifiques afin de réduire les délais de développement donc optimiser les coûts.

A travers l'ensemble de ces activités j'ai été en contact avec des responsables de différents services au sein de la société afin obtenir les informations concernant les cartes électroniques, la mécanique, le plan de câblage du lecteur. Ainsi, j'ai pu discuter avec eux et d'apporter des idées nouvelles afin d'améliorer les performances du lecteur. J'ai pu développer des qualités relationnelles avec ces personnes et c'est là que je me suis rendu compte que la communication est très importante pour faire passer ses idées et pour se faire entendre.

En tant que technicien supérieur, j'ai toujours exercé cette fonction au même titre qu'un ingénieur au sein du service avec les mêmes responsabilités mais sans les avantages. J'ai aussi à plusieurs reprises demandé à avoir un statut de cadre mais la réponse a été toujours négative. Pour remédier à cette situation, j'ai donc décidé de suivre la formation FIM en vue d'acquérir de nouvelles connaissances, d'être plus performant, d'assumer plus de responsabilités et surtout d'avoir un profil de cadre afin d'évoluer dans l'avenir vers un poste de chef de projet ou d'encadrement.