

Matthias MEUSBURGER  
Année 2000-2001  
IUT INFORMATIQUE BELFORT

# Rapport de stage



Responsable de stage :  
David LAIYMANI

# I. REMERCIEMENTS

Tout d'abord, je tiens à remercier l'équipe de MSG Software et particulièrement Steve Gontard pour m'avoir permis d'effectuer ce stage, ainsi que Didier Daubard pour son aide en Visual Basic.

De plus, je remercie également l'équipe pédagogique de l'IUT informatique de Belfort pour la formation qu'elle m'a apporté durant ces deux années, et particulièrement mon maître de stage, David Laiymani.

## II. SOMMAIRE

I.	Remerciements.....	2
II.	Sommaire .....	3
III.	Introduction.....	4
IV.	Présentation de l'entreprise.....	5
V.	Présentation du sujet .....	8
VI.	Travail accompli .....	11
VII.	Difficultés rencontrées .....	17
VIII.	Compétences utilisées.....	19
IX.	Compétences acquises .....	21
X.	Conclusion .....	23
XI.	Lexique .....	25

### III. INTRODUCTION

La formation suivie à l'IUT Informatique de Belfort m'a tout naturellement conduit, en fin de parcours, à prendre connaissance de la réalité du métier d'analyste-programmeur par le biais d'un stage de fin d'études.

J'ai choisi d'effectuer ce stage chez MSG Software, société belfortaine spécialisée dans la gestion électronique de documents.

Le travail que j'ai accompli au sein de cette société m'a fait découvrir des domaines d'activités ainsi qu'un langage de programmation qui m'étaient inconnus jusqu'alors.

De ce fait, lorsque les compétences acquises à l'IUT se sont révélées insuffisantes pour résoudre un problème particulier, les difficultés rencontrées, une fois surmontées, m'ont permis d'enrichir mes connaissances en complément à la formation théorique.

## IV. PRESENTATION DE L'ENTREPRISE

### IV.1 Présentation de MSG Software



La Société belfortaine MSG Software est née en novembre 1992 de la rencontre de 3 personnes issues du milieu informatique et du milieu industriel.

Ses fondateurs sont Jean-Claude MONNIER, Jean-Luc SCHMIDLIN et Steve GONTARD.

Bien que sa vocation de départ ait été la Gestion Electronique de l'Information et des Documents Existants (la G.E.I.D.E), la société est aujourd'hui une PME d'une trentaine de personnes dont les activités se sont diversifiées.

En effet, en novembre 1999, la société GED Software est créée, filiale dédiée aux nouveaux produits de GEIDE et au tiers archivage légal.



De même, octobre 2000 voit MSG Software donner naissance à MEDSOFT, filiale spécialisée dans les produits à destination du monde médical.

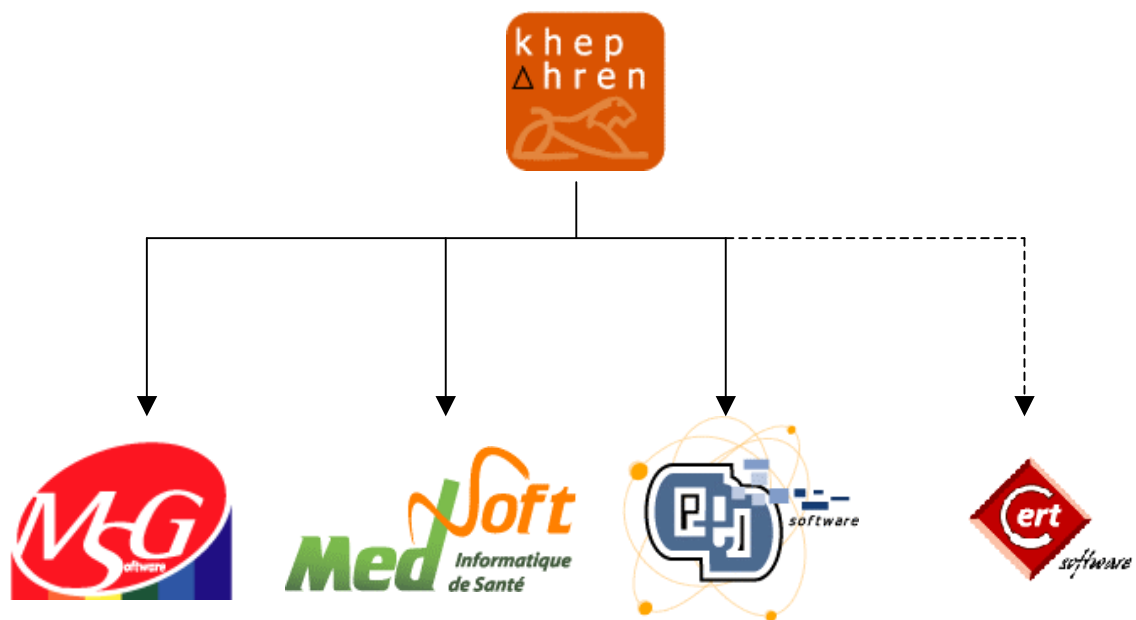
Finalement, dans un souci d'unification, MSG Software, GED Software, et MEDSoft sont regroupés en février 2001 au sein de la holding Kephren.



D'autre part, la société connaît une forte croissance, puisqu'une vingtaine de personnes doivent être embauchées d'ici la fin de l'été.



De plus, Cert Software, filiale spécialisée dans la certification de documents électroniques, doit être créée dans les prochains mois.



Structure de la société

## IV.2 Présentation des principaux produits MSG Software

La plupart des produits MSG Software sont écrits en Visual Basic et développés sous un environnement Windows pour un environnement Windows. Les applications présentées ici relèvent de la GEIDE, activité principale de MSG Software.

- **PUCCINI**

Puccini est l'application d'archivage et de consultation des documents. C'est un ensemble de programmes paramétrables permettant l'acquisition de tout types de documents.

- **ROSSINI**

Rossini est un ensemble d'applications paramétrables de gestion et d'administration des documents stockés.

- **VIVALDI (COLD)**

Vivaldi est une application d'archivage automatique de documents depuis des systèmes informatiques centraux avec contrôle de niveau de sauvegarde.

- **VERDI**

Verdi est une application qui permet de manipuler des documents de la GEIDE à partir de systèmes informatiques centraux existants (gestion commerciale, de salaire et de production...)

Cette interface reprend toutes les fonctionnalités du progiciel Puccini.

- **PUCCINI - WEB**

Puccini – Web est l'exception en terme de développement. Cette application est écrite en Java. Ses principaux atouts sont de vouloir proposer des services simplifiés de Puccini dans un environnement Intranet ou Extranet. Elle fonctionne sous Navigateur Internet et est donc totalement indépendante du système d'exploitation utilisé.

Cette application très prometteuse est fonctionnelle depuis peu.

## V. PRESENTATION DU SUJET

Affecté à MEDSoft, mon travail durant ce stage a été orienté vers l'édition d'états pour le logiciel STEPPO. Pour définir clairement la nature de cette tâche, il convient avant toutes choses d'expliciter la notion d'édition d'état, de présenter le logiciel STEPPO et de montrer l'importance du premier au sein du second.

### V.1 L'édition d'états

Editer un **état**<sup>1</sup> consiste à se connecter à une source de données, telle une qu'une base de données ou un **jeu d'enregistrements** pour y récupérer des informations que l'on va organiser et mettre en forme en vue d'une impression papier.

Ainsi, on peut présenter ces données selon un angle voulu, et faire apparaître des caractéristiques sous-jacentes qui n'auraient pas été visibles en consultant la source de données elle-même. Et bien évidemment, cela permet d'améliorer sensiblement l'esthétique de la présentation.

On utilise pour cela des **générateurs d'états** tels que *Access* de *Microsoft*, *Crystal Reports* de *Seagate Software*, *Report Maker* de *Synactis*, ou encore *QuickReports* de *QuSoft*.

---

<sup>1</sup> Les termes définis dans le lexique sont présentés en bleu lors de leur première apparition dans le rapport. Un aperçu concret d'état est visible dans l'annexe A partie 2 (page 5).

## V.2 Le logiciel STEPRO

Stépro est un logiciel de gestion de la stérilisation<sup>1</sup> développé par Medsoft, filiale de MSG Software, et résulte de la conversion d'une application existante sous Microsoft Access, STE-GAO.

Stépro a pour but d'optimiser la production et de gérer les dépenses concernant les outils médicaux dans les hôpitaux et les cliniques, mais surtout de mettre en place une traçabilité des instruments utilisés. En effet, une partie non négligeable du matériel utilisé est recyclable. Suite à l'apparition de certaines contaminations comme la maladie de Creutzfeldt-Jakob, dont le prion responsable de sa propagation nécessite un traitement très particulier, la traçabilité des outils est devenue indispensable. C'est dans le but de simplifier mais aussi de perfectionner ce procédé de suivi des produits que Stépro a été développé.

## V.3 L'importance de l'édition d'états dans STEPRO

L'édition d'états est un aspect fondamental de Stépro. En effet, le support papier est le seul moyen d'échanger des données entre les sites informatisés et ceux qui ne le sont pas, sachant que le cycle de vie d'un instrument peut ne pas être interne à un seul établissement. C'est le cas par exemple en ce qui concerne le matériel très spécifique, comme celui utilisé pour poser une prothèse, et qui est prêté par le fabricant. Ce matériel « voyagera » alors de site en site durant son cycle de vie, et il faut pouvoir conserver une trace ses différentes utilisations. Ainsi, le support papier permet d'assurer la traçabilité d'un article quelle que soit la situation.

De plus, le procédé d'édition d'états va servir à l'impression d'étiquettes codes barres utilisées pour identifier les instruments lors du cycle de stérilisation.

Mon travail sur l'application Stépro a concerné le module catalogue. Cette partie du logiciel permet de consulter les différents produits et articles médicaux, ainsi que leurs caractéristiques : cycle de vie, date de péremption, usage, quantité...

---

<sup>1</sup> Un aperçu de l'implantation de l'informatique au sein du cycle de stérilisation est donné dans l'annexe A partie 4 (page 11).

#### V.4 Nécessité de déterminer la manière d'éditer des états dans STEPPO

Dans l'ancienne application, STE-GAO, la conception et l'édition des états s'effectuaient directement sous Access. Cependant, Stépro étant développé en Visual Basic, il a fallu choisir un logiciel d'édition d'états externe qui soit compatible avec cet environnement de programmation. Cela a constitué mon premier travail : utiliser deux logiciels de génération d'état préalablement choisis par l'entreprise : *Crystal Reports* de *Seagate Software* et *Report Maker* de *Synactis*, puis rédiger un rapport comparatif décrivant les avantages et inconvénients de chacun, afin qu'une décision soit prise quant au logiciel à utiliser.

A l'issue de ce travail d'environ deux semaines<sup>1</sup>, *Crystal Reports* s'est imposé comme étant le logiciel le plus adapté. Il a ensuite fallu appréhender l'utilisation de ce logiciel afin de constituer un manuel d'utilisation, mais aussi un manuel d'intégration à Visual Basic. Cette intégration étant, suivant les cas de figures, plus ou moins complexe, j'ai également mis au point une application de démonstration en Visual Basic.

---

<sup>1</sup> Un planning complet des tâches effectuées durant le stage est disponible dans l'annexe A partie 1 (page 4)

## VI. TRAVAIL ACCOMPLI

### VI.1 Comparatif Crystal Reports / Report Maker

Mon étude comparative<sup>1</sup> de ces deux logiciels s'est scindée en deux parties principales :

- L'utilisation du logiciel en lui-même.
- L'intégration des fonctions d'édition d'états au sein de Visual Basic.

#### VI.1.a L'utilisation des logiciels en eux-mêmes

Voici les critères qui ont guidé mon étude de ces deux logiciels :

- 1) L'installation et le paramétrage du logiciel
- 2) La qualité et l'ergonomie de l'interface graphique
- 3) L'ajout d'une **table** à l'état
- 4) Le formatage des **champs**
- 5) La création d'une **requête**
- 6) L'insertion et la consultation des champs au sein d'un état
- 7) Les champs spéciaux
- 8) Le système de sauvegarde des fichiers
- 9) La stabilité du logiciel

Bien évidemment, certains de ces critères ont une importance bien plus grande que d'autres, c'est d'ailleurs pourquoi dans le document comparatif, les notes attribuées dans le tableau récapitulatif tiennent compte et de la qualité de l'application sur un critère donné et de l'importance du critère en question. Cependant, il m'a semblé primordial que tous ces critères, même ceux qui ne relèvent pas forcément d'un point capital de l'édition d'états, soient présents dans mon étude, car la facilité et même le plaisir d'utilisation d'un logiciel au quotidien sont souvent conditionnés par de petits détails de conception, qui font de son maniement une sinécure ou un pensum.

---

<sup>1</sup> disponible dans l'annexe B : « Comparatif Crystal Reports / Report Maker » (page 13).

### VI.1.b L'intégration des fonctions d'édition d'états au sein de Visual Basic

L'intégration des fonctions d'édition d'états au sein de Visual Basic constitue le point le plus décisif concernant le choix du logiciel qui sera adopté par l'entreprise. En effet, bien que la création des états se fasse via le logiciel de génération d'états, leur édition se fera inmanquablement dans Visual Basic, de sorte que l'utilisateur final de Stépro ne soit pas dans l'obligation de posséder une licence d'utilisation du générateur, à moins qu'il ne souhaite créer lui-même ses propres états.

Pour pouvoir éditer un état au sein de Visual Basic, l'application génératrice d'état fournit des composants Visual Basic qui permettront, via des boîtes de dialogue ou au moyen de lignes de code, d'ouvrir les fichiers contenant les états à éditer. L'édition peut alors se faire indépendamment du logiciel générateur.

Pour chacun des deux logiciels, ce comparatif présente :

- L'intégration brute d'un rapport
- L'intégration avec passage d'un jeu d'enregistrement au rapport
- La qualité de l'aide fournie
- Les codes source et explications détaillées des deux types d'intégration

Pour des raisons de clarté, seules des appréciations d'ordre général seront présentes ici concernant les différentes techniques d'intégration d'un rapport. En effet, ces sujets seront approfondis dans le rapport technique.

#### VI.1.b.i L'intégration brute d'un rapport

Le terme « intégration brute » désigne ici le fait d'insérer dans une application Visual Basic un rapport tel qu'il a été conçu dans le logiciel propriétaire de ce rapport, sans y apporter aucune modification.

Il s'agit de la manière la plus basique d'intégrer un rapport à une application. Sur ce point, Report Maker est plus convivial que Crystal Reports, puisqu'il permet d'effectuer cette tâche sans utiliser de code. Tout se fait automatiquement, via les composants fournis par le logiciel. Avec Crystal Reports, la saisie de quelques lignes de code est inévitable.

### VI.1.b.ii L'intégration avec passage d'un jeu d'enregistrements au rapport

Cependant, l'intégration brute ne sera que très rarement, voire jamais, utilisée au sein de STEPPO. En effet, l'édition de tous les articles présents dans une base de données ne correspond pas aux réalités des besoins des futurs clients de l'application. De manière générale dans STEPPO, l'utilisateur peut appliquer des tris et des filtres sur les données qu'il souhaite éditer, et ce, avant l'édition de l'état. Les fonctions d'édition d'états du logiciel générateur doivent alors pouvoir moduler l'édition d'un état. Il est donc souhaitable de pouvoir passer directement à l'état un jeu d'enregistrements résultant des tris et filtres appliqués par l'utilisateur. L'état édité concernera alors uniquement les données voulues. C'est cette technique qui constitue le deuxième point : L'édition avec passage d'un jeu d'enregistrements au rapport.

C'est ce critère qui a surtout conditionné le choix du logiciel générateur. En effet, il s'est avéré impossible de réaliser ce type d'édition avec Report Maker. Néanmoins, l'aide du logiciel laisse entendre que cela est possible sous certaines conditions qui y sont tout juste mentionnées sans y être explicitées. Cela m'a amené à mener une étude sur un troisième critère : la qualité de l'aide disponible pour chacun des logiciels.

### VI.1.b.iii Qualité de l'aide

Si l'utilisation d'un générateur d'état peut se faire de manière relativement intuitive lorsqu'il s'agit de créer un état via l'interface graphique du logiciel, l'implémentation des états au sein d'un quelconque langage de programmation ne tolère plus d'à-peu-près.

En effet, les outils mis à disposition pour effectuer ce travail sont souvent complexes de par le grand nombre de cas de figures pouvant se présenter lors de l'édition d'un état.

Dès lors, une aide incomplète peut se révéler être un facteur bloquant pour le développement d'une application. Il est donc nécessaire que l'aide disponible couvre suffisamment de domaines, et présente une garantie d'évolution minimum, pour être viable à long terme.

Pour chacun des logiciels, l'aide à été évaluée selon trois points :

- L'aide fournie avec le logiciel : exhaustivité et clarté
- L'aide disponible sur Internet
- Les exemples fournis

Cependant, le support technique (courrier électronique – téléphone) n'a pu être pris en compte lors cette étude. En effet, les logiciels testés étaient tous deux en version d'évaluation.

Report Maker s'est révélé être très pauvre en explications détaillées. De plus, le site Internet de l'entreprise n'est d'aucun secours en cas de difficulté et les exemples fournis ne décrivent que les procédures basiques d'intégration à Visual Basic

En ce qui concerne Crystal Reports, l'aide fournie est très importante, bien que seule la documentation concernant l'utilisation du logiciel soit en français. En effet, l'aide à destination du développeur est entièrement en anglais. Les documents additionnels proposés par l'éditeur sur son site Internet sont également en anglais, mais il est possible de trouver de l'aide sur n'importe quelle fonctionnalité du logiciel. Cependant, j'ai dû utiliser autant les aides fournies avec le logiciel que celles disponibles sur le site Internet et cela induit un certain manque d'organisation dans les documents. L'aide aurait mérité d'être centralisée dans les fichiers livrés sur le CD-ROM du logiciel.

Au terme de l'étude comparative, bien que Report Maker de Synactis se soit avéré plus abordable concernant la prise en main pour un utilisateur novice, Crystal Reports de Seagate Software s'est imposé comme étant le logiciel le plus complet.

Cependant, les différents points abordés lors du comparatif des deux logiciels ne suffiraient pas à guider un utilisateur cherchant à effectuer une utilisation avancée de Crystal Reports. Ainsi, il a fallu d'une part expliciter le maniement du logiciel en lui-même et développer le fonctionnement de ses outils d'intégration à Visual Basic.

## VI.2 Documentation interne concernant Crystal Reports

### VI.2.a Manuel d'utilisation

Le manuel d'utilisation<sup>1</sup> de Crystal Reports doit permettre une prise en main rapide et efficace pour que n'importe quel utilisateur puisse créer un état de base. Sans être extrêmement détaillé, puisque l'interface du logiciel reste tout de même relativement classique, il doit permettre de créer des états simples contenant **groupes** et **sous-états**. La difficulté augmente au fur et à mesure des points abordés.

La première partie permet de créer un état, avec le détail de la connexion à la base de données et la présentation des différentes fenêtres entrant en jeu dans la procédure de création.

La seconde partie présente les différentes **sections** qui composent un état ainsi que leur fonction.

La troisième partie montre l'intérêt de créer un état contenant des groupes, ainsi que la manière d'y parvenir.

La quatrième partie montre dans quels cas créer un sous-état, ainsi que la manière d'y parvenir.

Finalement, la dernière partie explique comment utiliser un champ de formule pour insérer un code-barre dans l'état.

### VI.2.b Manuel d'intégration à Visual Basic

Le manuel d'intégration<sup>2</sup> à Visual Basic reprend les méthodes présentes dans la partie du comparatif concernant la programmation en approfondissant et en présentant d'autres notions intervenant dans l'édition d'états élaborée. Ainsi, les méthodes pour insérer un champ dynamique, c'est-à-dire dont la valeur est déterminée au moment de l'exécution de l'application, viennent compléter les méthodes d'intégration brute et d'intégration avec passage d'un jeu d'enregistrement.

Cependant, la rédaction de manuels peut ne pas être suffisante pour comprendre toutes les subtilités d'un procédé de programmation.

---

<sup>1</sup> Disponible dans l'annexe C : « Documentation interne concernant Crystal Reports » partie 1 (page 39)

<sup>2</sup> Disponible dans l'annexe C : « Documentation interne concernant Crystal Reports » partie 2 (page 47)

### VI.3 Réalisation d'une application-exemple

Afin de concrétiser la documentation sur l'intégration d'états au sein de Visual Basic, j'ai réalisé une application, écrite dans ce même langage, et qui reprend tous les cas de figure que MSG Software pourrait rencontrer lors de la suite du développement de STEPPO. Ainsi, la personne qui reprendra mon travail aura un aspect pratique des différentes méthodes qu'il est possible d'utiliser, tout en ayant conscience de facteurs qu'il est impossible de juger sur papier tels que les temps de traitement des informations ou les temps d'accès à la base de données. Elle pourra de cette manière plus facilement décider de la technique à adopter pour un cas de figure donné en tenant compte de ces paramètres.

Ainsi, mon travail durant ce stage a plus été orienté vers la documentation que vers la programmation. En dépit de la nature peu habituelle de ma tâche, je pense avoir correctement atteint les objectifs fixés, mais certaines difficultés se sont présentées à moi lors de la réalisation de ces travaux. Cependant, une fois surmontés, ces obstacles m'ont permis d'acquérir de nouvelles connaissances.

## VII. DIFFICULTES RENCONTREES

### VII.1 L'environnement de développement utilisé

Formé par l'IUT à travailler sous UNIX/LINUX, le passage au développement sous WINDOWS a nécessité un léger temps d'adaptation. Cependant, le changement le plus notable a été de passer des langages de programmation enseignés à l'IUT à Visual Basic, langage de programmation événementiel. En effet, ce concept m'était totalement étranger à mon arrivée chez MSG Software, et le fait d'utiliser un environnement de développement graphique a suscité chez moi quelques surprises. Sous Visual Basic, il n'est pas nécessaire de structurer sous forme de code le squelette du programme, puisque les différents objets (feuilles, boutons, liste déroulante, ...) sont générés par glisser-déposer à partir d'une barre d'outil.

De plus, habitué à développer avec un simple éditeur de texte, j'étais totalement étranger aux différents outils d'aide à la programmation, tels que les points d'arrêts, les variables espions, et l'interrogation de variables au cours de l'exécution du programme. Cependant, en dépit du temps d'adaptation nécessaire à ces changements, la prise en main de ces outils s'est relativement bien passée. En effet, ceux-ci ne constituent que des « à-côtés » à la programmation, et, dans le sens où ceux-ci ne sont justement pas vitaux pour le développement d'une application, la difficulté n'a pas été insurmontable.

## VII.2 Différences entre les cas d'écoles et les cas pratiques

La différence entre les cas étudiés à l'IUT et ceux qui se présentent dans la pratique m'a également surpris. En effet, lors de la création des états destinés à établir le comparatif entre les deux logiciels générateurs, j'ai eu à trouver les correspondances concernant les différentes tables et champs entre l'ancienne base de données de l'application STE-GAO fonctionnant sous Access, et la nouvelle base de données de STEPRO fonctionnant sous Oracle. Ce ne fût pas chose aisée, puisque ces bases de données sont énormes en comparaison des cas étudiés à l'IUT. Alors que la base du projet de base de données effectué à l'IUT comptait une vingtaine de tables, celle de la seule partie catalogue de STE-GAO en compte plus d'une soixantaine.

De même, en ce qui concerne la complexité du code SQL à construire, je me suis rendu compte qu'à la différence de l'IUT où le code était très complexe et portait sur une faible quantité d'information, dans la pratique le code s'est révélé peu complexe mais portant sur un grand nombre d'informations.

Par ailleurs, en dehors des problèmes dus à l'adaption à un nouvel environnement de travail, il m'a été difficile sur certains points de mettre en pratique les connaissances acquises à l'IUT.

## VII.3 Mise en place de la méthode MERISE des tests et recettes

La méthode des tests et recettes de MERISE permet de d'évaluer l'adéquation fonctionnelle, organisationnelle et technologique d'un logiciel, d'un composant logiciel ou d'un système informatisé.

J'ai rencontré certaines difficultés à appliquer cette méthode à l'étude comparative des logiciels générateurs d'états que j'ai menée pendant ce stage. En effet, la méthode en question traite surtout de l'étude des logiciels développés en interne, dans un but de correction et de mise au point, alors que les logiciels testés dans mon travail étaient déjà des logiciels finalisés.

Cependant, j'ai pu effectuer un test de stress sur ces logiciels pour déterminer quelle utilisation dépassait leur limites.

## VIII. COMPETENCES UTILISEES

En dehors des connaissances générales, qui servent dans tout travail relatif à l'informatique, certaines connaissances plus spécifiques acquises à l'IUT m'ont permis d'atteindre les objectifs de mon stage.

### VIII.1 Le langage SQL

La connaissance du langage SQL a été fort pratique lors du test des générateurs d'états. En effet, cela m'a permis de juger personnellement de la qualité du code SQL généré par ces applications, et donc, de leur efficacité.

De plus, la connaissance de ce langage m'a permis d'appliquer des opérations complexes sur ces logiciels, pouvant ainsi effectuer le test de stress recommandé par la méthode MERISE. De cette manière, j'ai pu déterminer quelle utilisation dépassait les limites du logiciel<sup>1</sup>.

### VIII.2 La méthode des tests et recettes de MERISE

Bien que l'utilisation de cette méthode se soit avérée délicate en ce qui concerne le comparatif des logiciels générateurs d'états, il m'a été possible de la mettre en œuvre sur certains points lors de l'étude de l'intégration d'états à Visual Basic.

En effet, dans ce contexte, il était relativement aisé de substituer l'utilisation des outils fournis avec les logiciels générateurs d'états au développement d'une application interne. Ainsi, j'ai répondu aux attentes exigées par la méthode au premier et au troisième niveau de conception.

Premièrement, le besoin de satisfaire la demande du client (premier niveau de conception), en l'occurrence MSG Software. En effet, l'édition brute d'un état ne convenant pas au chef de projet STEPPO, nous avons convenu qu'il fallait une méthode plus souple et plus rapide, et c'est ce qui m'a poussé à étudier les autres types d'intégration d'état. Ainsi, la relation client/fournisseur revêtait un aspect primordial, conformément aux instructions de la méthode des tests et recettes.

---

<sup>1</sup> voir annexe B partie 1 section IV paragraphes 3 et 5 (pages 17 et 18)

Ensuite, le fait de mettre en place un système de mesure (troisième niveau de conception) a été concrétisé par le développement de l'application-exemple. En dépit du fait que cette application ne renvoie pas la valeur numérique correspondant au temps de traitement, elle permet de juger des différences significatives de réponse des différentes méthodes utilisées. Ainsi, le développeur, c'est-à-dire en fait l'utilisateur de ces méthodes, pourra juger rapidement de l'adéquation des temps de traitements aux besoins de la situation.

Par ailleurs, en dehors de l'utilisation des compétences enseignées à l'IUT, certains des problèmes qui se sont posés à moi lors de ce stage m'ont permis de d'acquérir de nouvelles connaissances.

## IX. COMPETENCES ACQUISES

### IX.1 Autonomie

Pendant les deux années d'enseignement à l'IUT, nous avons appris à travailler seul ou en groupe, au travers des différents TP et projets. Cependant, durant ces travaux, il était toujours possible de demander conseil aux autres étudiants ou aux professeurs en cas de besoin. De plus, l'IUT nous a toujours donné une méthodologie relativement précise concernant un sujet donné, alors qu'étant seul à travailler sur mon sujet de stage, et opérant sur des logiciels que personne dans l'entreprise ne connaissait, j'ai dû apprendre à travailler de manière réellement autonome, et à donner une orientation par moi-même choisie aux différents résultats des études que j'ai menées.

### IX.2 Esprit de synthèse

Mon travail de comparaison des deux logiciels et de documentation de Crystal Reports s'est avéré être un travail d'épuration. En effet, la documentation livrée avec le logiciel représente plus de 2000 pages, dont environ 800 pages en français et 1400 pages en anglais ! Malgré ce nombre de pages imposant, la documentation fournie n'est pas exhaustive. En effet, pour couvrir tous les cas qui m'intéressaient, j'ai téléchargé en plus sur le site Internet de l'éditeur plusieurs centaines de pages supplémentaires et quelques application exemples, le tout en anglais. La plupart des réponses aux questions qui me sont apparues lors de problèmes inexplicables ont été trouvées à force de parcourir les FAQ (Frequently Asked Questions – Questions souvent posées) de ce site. Ainsi, bien que dans l'absolu, je n'aie pas tapé énormément de code, mon travail a été de débroussailler le terrain pour les programmeurs qui participeront à ce projet après moi.

Il faut savoir que Crystal Reports est un outil relativement ancien puisque la première version date de 1991, et les outils d'intégration aux différents langages de programmation sont extrêmement nombreux. De plus, pour un même langage de programmation (ici Visual Basic), il existe plusieurs méthodes pour effectuer la même tâche. Cela provient en partie du fait que les différents outils proposés par Seagate Software ont évolué au fil des années, ont été remplacés par d'autres outils plus performants bien que les anciennes versions aient été conservées pour des raisons de compatibilité. D'autre part, certains outils ont été développés en parallèle, et s'ils permettent d'effectuer les mêmes actions, ils présentent des

caractéristiques de programmation différentes. Au final, pour implémenter un cas précis d'édition, j'ai dû bien souvent tester plusieurs méthodes avant de trouver celle qui convenait le mieux au contexte de STEPPO.

Ainsi, de nombreuses heures ont été passées à supprimer les solutions inadéquates et j'ai dû faire preuve d'un certain discernement pour proposer, dans les manuels à l'intention de MSG Software, les solutions les plus adaptées et les plus évidentes à mettre en œuvre.

## X. CONCLUSION

Au terme de ce stage, l'expérience s'est révélée positive. En effet, la qualité pédagogique de cette incursion dans le monde du travail est indéniable. Elle m'a permis de quitter le domaine théorique de l'informatique pour découvrir quelles sont les réalités de l'entreprise. Ainsi, j'ai pu, suivant la nature des tâches à effectuer, adapter les connaissances et méthodes acquises à l'IUT pour faire face aux problématiques nouvelles. De ce fait, les difficultés surmontées ne sont que des tremplins vers l'acquisition de nouvelles compétences.

En ce qui concerne le travail en lui-même, je pense avoir atteint les objectifs fixés, puisque la personne qui continuera à développer l'application Stépro disposera d'un manuel qui, je l'espère, lui permettra d'aller à l'essentiel en ce qui concerne l'édition d'états.

## TABLE DES MATIERES :

I.	Remerciements.....	2
II.	Sommaire .....	3
III.	Introduction.....	4
IV.	Présentation de l'entreprise.....	5
	IV.1 Présentation de MSG Software .....	5
	IV.2 Présentation des principaux produits MSG Software .....	7
V.	Présentation du sujet .....	8
	V.1 L'édition d'états .....	8
	V.2 Le logiciel STEPPO.....	9
	V.3 L'importance de l'édition d'états dans STEPPO.....	9
	V.4 Nécessité de déterminer la manière d'éditer des états dans STEPPO.....	10
VI.	Travail accompli .....	11
	VI.1 Comparatif Crystal Reports / Report Maker .....	11
	VI.1.a L'utilisation des logiciels en eux-mêmes.....	11
	VI.1.b L'intégration des fonctions d'édition d'états au sein de Visual Basic.....	12
	VI.2 Documentation interne concernant Crystal Reports .....	15
	VI.2.a Manuel d'utilisation.....	15
	VI.2.b Manuel d'intégration à Visual Basic .....	15
	VI.3 Réalisation d'une application-exemple .....	16
VII.	Difficultés rencontrées .....	17
	VII.1 L'environnement de développement utilisé .....	17
	VII.2 Différences entre les cas d'écoles et les cas pratiques .....	18
	VII.3 Mise en place de la méthode MERISE des tests et recettes.....	18
VIII.	Compétences utilisées.....	19
	VIII.1 Le langage SQL.....	19
	VIII.2 La méthode des tests et recettes de MERISE.....	19
IX.	Compétences acquises .....	21
	IX.1 Autonomie .....	21
	IX.2 Esprit de synthèse.....	21
X.	Conclusion .....	23
XI.	Lexique .....	25

## XI. LEXIQUE

**champ** : zone d'un état destinée à recevoir des données pour les afficher ensuite.

**état** : document qui présente des données mises en forme. L'édition d'un état permet de présenter des données de manière esthétique, mais plus important encore, les données à utiliser peuvent être soumises à certaines conditions.

*Exemples d'états :*

présenter les caractéristiques de tous les produits d'une certaine catégorie.

présenter le détail de ventes de l'année  $n$  à l'année  $m$ .

imprimer la liste des clients ayant acheté pour plus de  $n$  francs à l'entreprise.

**fichier rapport** : fichier contenant les informations nécessaires à l'édition d'un état, telles que celles concernant la mise en forme ou le type de données à utiliser.

**générateur d'états** : application qui permet de créer un **fichier rapport**.

*Exemples : Access de Microsoft, Crystal Reports de Seagate Software.*

**groupe** : comme son nom l'indique, un groupe permet de réunir différentes informations.

*Exemple* : un produit est composé de plusieurs articles. Si l'on veut afficher les noms des articles pour chaque produit, on peut le faire au moyen de groupes.

**jeu d'enregistrements** : forme sous laquelle une base de données renvoie les informations demandées.

**Oracle** : Système de Gestion de Bases de Données

*Autres exemples : Access, MySQL*

**requête** : constitue la demande d'information à la base de données. Dans les générateurs d'états, une requête peut-être créée soit de manière graphique soit directement en SQL.

**section** : partie de l'état jouant un rôle précis au moment de l'édition.

*Exemple* : la section « Pied de page de l'état » n'apparaîtra qu'à la fin de la dernière page de l'état.

**sous-état** : état inséré à l'intérieur d'un autre. Son mode de création est identique à celui d'un état standard, bien qu'il ne puisse avoir d'existence propre. Le plus souvent, le sous-état doit être mis en relation avec l'état principal, appelé alors état-maître.

**SQL** : Simple Query Language. Langage qui permet d'interroger une base de données.

**table** : Ensemble de données concernant une même rubrique au sein d'une base de données.

*Exemple* : la table « Produit » contiendra les numéros des produits, les noms des produits, les quantités disponibles des produits, etc...

**test de stress** : dans le cadre de la méthode des tests et recettes de MERISE, ce test vise à mesurer le comportement d'un logiciel lorsque celui-ci atteint ses limites.