Evolution de la solution SAP mise en place chez un client suite à son développement

Tuteur pédagogique
Caroline Thierry

Maitre de stage
Philipe Joubert
Lionel Arzur

Etudiant
Armand Fruitet

M2 ISMAG - Année Universitaire 2013/2014

Remerciements

Je tiens à remercier Messieurs André BAVEUX et Philippe LETELLIER de m'avoir accueilli au sein de leur structure.
Je remercie également Lionel ARZUR, responsable de la division technique, Philippe JOUBERT, mon maitre de stage, ainsi que l'ensemble de l'équipe d'Augusta Reeves pour m'avoir accompagné et apporté l'aide dont j'ai pu nécessiter durant ces six mois.

Je veux aussi remercier mon tuteur Caroline THIERRY pour sa disponibilité ainsi que toute l'équipe du corps enseignant qui m'a suivi pendant mes années de formation.

Contenu

Remerciements 1

Introduction 3

I) Contexte 4

A) Augusta Reeves Group (1) 4

1. Historique 4

2. Fonctionnement 4

3. Activités 6

4. Méthodologie d'Augusta Reeves 6

B) Outil utilisé 8

1. SAP 8

a) Le progiciel SAP (2) 8

b) Caractéristiques de SAP R/3 (2) 8

c) Les modules SAP R/3 9

d) Le langage ABAP/4 11

e) Navigation dans SAP 11

f) Informations sur le code ABAP 12

g) Exemples de réalisations avec le code ABAP (3) 12

II) Travaux réalisés 14

A) Création de BAPI pour l'entreprise ALTERBURO 14

1. Présentation du contexte 14

2. Création des BAPI 15

3. Taches réalisées 16

a) Sélection des villes en fonction du code postal 16

b) Sélection des interlocuteurs de niveau supérieur à celui renseigné 17

B) Amélioration du système déjà mis en place chez Flammarion suite à son évolution 21

1. Présentation du contexte 21

2. Taches réalisées 22

a) L'office 22

b) La fiche article 24

c) La pyramide éditoriale 27

Conclusion 33

Bibliographie 34

Introduction

Dans le cadre de mon Master 2 ISMAG (Informatique, Statistiques, Mathématiques appliqués à la Gestion de production), j'ai réalisé un stage de fin d'étude de 6 mois.
Parmi les demandes que j'ai faites, j'ai eu 2 réponses positives dont celle d'Augusta Reeves que j'ai choisie, dont le Siège se situe à Paris.

Ce stage de fin d'études a pour but d'appliquer les connaissances apprises, lors de la formation ISMAG, dans le milieu professionnel. Il permet aussi d'appréhender de nouvelles compétences et méthodologies de travail.

Durant ce stage, j'ai été amené à me familiariser avec le monde SAP puis à travailler avec ce dernier. Bien qu'ayant eu des cours sur les ERP ainsi que de brèves présentations sur les côtés techniques puis fonctionnels du logiciel par des professionnels il m'a fallu du temps pour mieux comprendre le fonctionnement, les capacités et possibilités offertes par SAP.
Cela est très enrichissant professionnellement car ce logiciel est leader dans son domaine, reconnu mondialement et offre tellement de possibilités qu'on apprend tous les jours.
Mon travail a consisté à me former sur SAP, plus particulièrement son langage de programmation ABAP durant le premier mois. J'ai ensuite continué ma formation en travaillant sur un projet durant deux mois. Enfin je suis passé sur un projet durant les 3 derniers mois de mon stage chez un client.

Dans un premier temps je présenterai l'entreprise Augusta Reeves dans laquelle j'ai effectué mon stage ainsi que le logiciel que j'ai utilisé. Ensuite j'expliquerai en quoi a consisté mon travail lors de la seconde partie de ma formation. Enfin je mettrai en avant mon travail et le projet dans lequel j'ai été affecté lors de mes trois derniers mois de stage, à savoir l'évolution de la solution déjà mise en place dans une entreprise suite à son développement.

I) Contexte

A) Augusta Reeves Group (1)

1. Historique

Le groupe Augusta Reeves Group, société informatique, débute son parcours en 1998, date de sa création par ses fondateurs Messieurs André BAVEUX et Philippe LE TELLIER. L'entreprise commence par proposer un service de conseils dans l'intégration de l'Enterprise Resource Planning (ERP) ou Progiciel de Gestion Intégré (PGI) SAP à une clientèle de professionnels.
Par la suite, l'entreprise élargit son service à l'intégration d'autres solutions SAP mais également d'autres technologies par le biais d'une croissance interne comme externe.

2. Fonctionnement

L'International Data Group (IDC), entreprise spécialisée dans l'étude des marchés des technologies de l'information, estime que le marché des logiciels professionnels a atteint 342 milliards de dollars en 2012 dans le monde soit une croissance en hausse de 3,6% par rapport à 2011 mais en perte de vitesse par rapport aux hausses des années précédentes. Bien qu'il ne fasse pas la course en tête, SAP connait la plus forte progression avec 5,1% de hausse à 5%. Par ailleurs, les applications de GRC (Gestion de la Relation Client) ou CRM (Customer Relationship Management) font partie de celles qui tirent les ventes du secteur avec 7% de hausse.

Dans le contexte d'évolution actuel le CA d'Augusta est logiquement en hausse sur ces dernières années, hausse qui est de 16% entre 2011 et 2012.

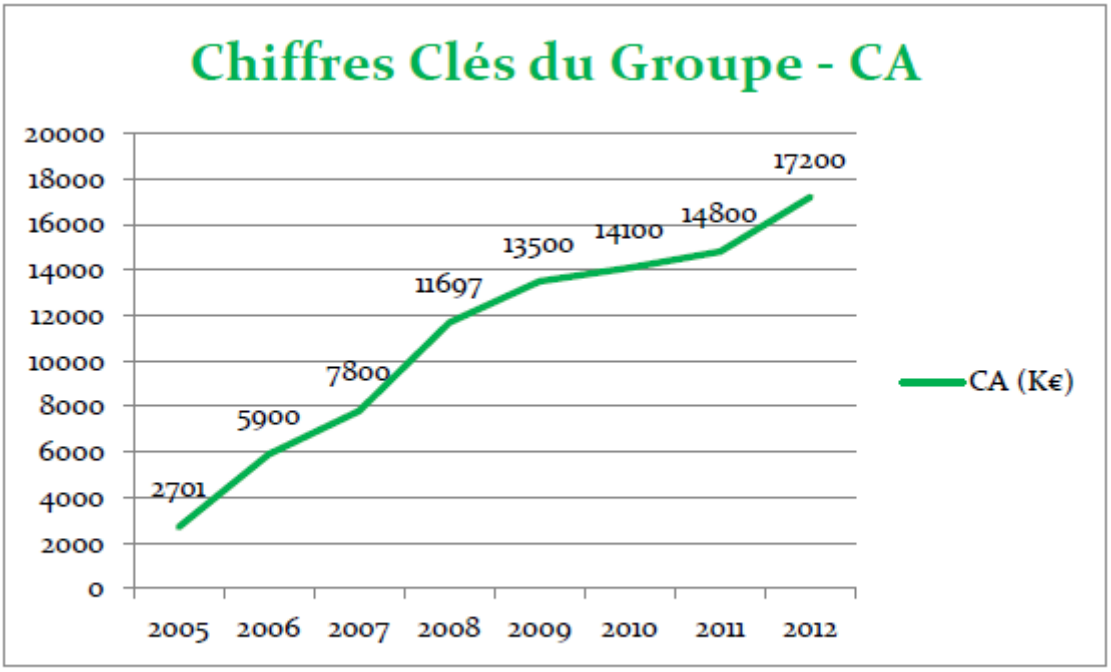


Figure 1 - Chiffres clés du Groupe - CA

Naturellement l'effectif de l'entreprise est également en hausse ces dernières années, 13% d'augmentation entre 2011 et 2012 par exemple.

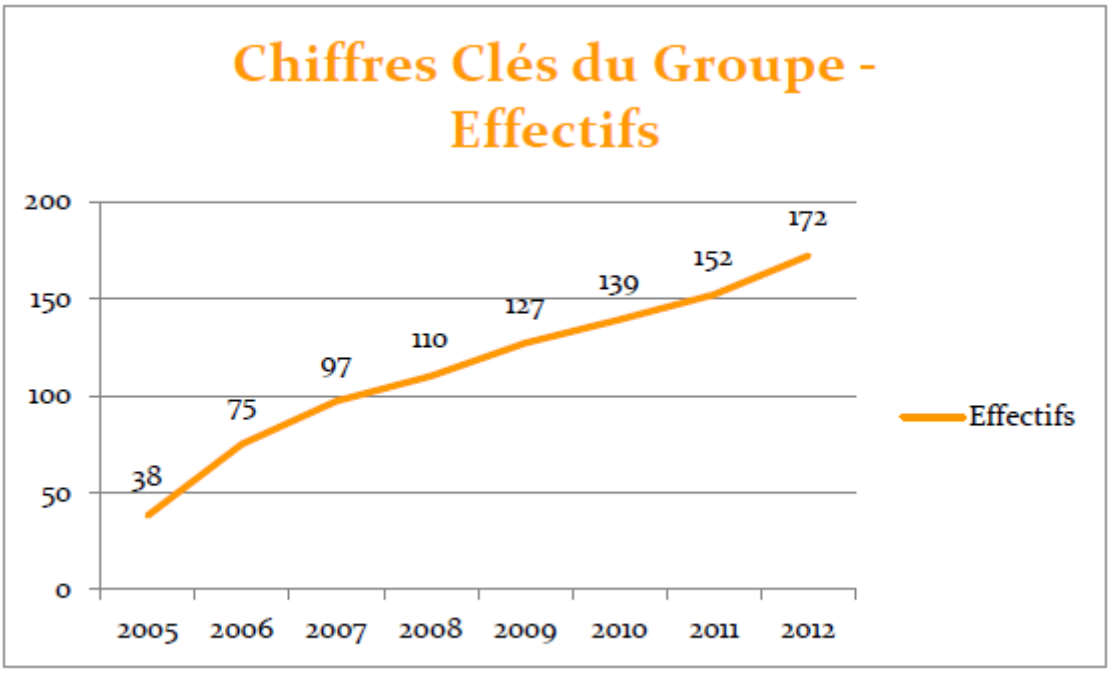


Figure 2 - Chiffres clés du Groupe - Effectifs

S.A.S. doté d'un capital de 5 367 085 euros détenu à 75% par les associés que sont Messieurs André BAVEUX, Eric GARAULT, Jean-Claude NIZARD et Philippe LE TELLIER, à 14% par la société de capital investissement Naxicap et à 11% par les salariés, le groupe est dirigé par Monsieur Philippe LE TELLIER.
Il se décompose entre les deux divisions que sont Augusta Reeves, qui se positionne sur l'intégration de solutions SAP, et X-WAN, qui a été créée par la suite et qui propose l'intégration d'autres technologies.
Bien qu'elles évoluent au sein d'environnements technologiques différents, elles visent toutes deux à satisfaire leurs clients.

3. Activités

Avec une offre d'activités complémentaires telles que l'intégration de solutions SAP mais aussi d'autres technologies, le groupe Augusta Reeves Group opère sur le marché informatique des professionnels.
Il dispose d'un vaste portefeuille de clients aux types d'activités variés parmi lesquels, entre autres, Air liquide, le Cemagref, la Cité des sciences, Cofiroute, la Cours des comptes, Saint-Gobain, Sanofi-Aventis, Aélia, Althea, Andra, la Lyonnaise des Eaux, le Cea, Chanel, l'Efs, Eni, Flammarion, la Fromagerie Bel, Inria, L'Oréal, Nestlé, Sodilac, Etam et Ingenico.
Doté de nombreuses certifications, avec entre autres celles de centre d'expertise SAP, il se pose en expert dont les compétences s'étendent de l'ERP à la Business Intelligence en passant par la CRM.
La mise en place de solutions de gestion depuis l'analyse des besoins jusqu'à la formation en passant par l'installation, le paramétrage et le recettage fait partie de son offre. Par ailleurs, en fonction des besoins de ses clients, la personnalisation des outils par le biais de développements spécifiques est possible si les besoins de ses clients sont trop spécifiques.
Cette dernière est l'étape que j'ai eu l'occasion d'approfondir plus particulièrement dans le cadre de mon affectation au sein de la cellule technique.

4. Méthodologie d'Augusta Reeves

Augusta Reeves a sa propre méthodologie pour aborder ses missions. En effet, la méthode classique, appelée méthode en V, consiste à recueillir le besoin du client, à réaliser ce besoin, à le tester puis à soumettre une proposition utilisable au client et à adapter celle-ci en fonction des rectifications qu'il exprime.
Cette méthode comporte ce qui est appelé « l'effet tunnel », c'est-à-dire qu'il peut s'écouler plusieurs mois sans qu'il y ait d'interaction entre le client et le consultant. Le client n'intervient qu'à deux reprises : au début, lorsqu'il exprime son besoin, et à la fin, lorsqu'il compare son

besoin avec la livraison. Cette méthode implique que la phase de recueil du besoin soit extrêmement précise et approfondie pour ne pas susciter de malentendu sur ce qu'attend le client.
Augusta Reeves a décidé d'annuler cet effet de tunnel en impliquant davantage le client dans le projet. Augusta Reeves accompagne donc le client dans son projet en n'étant pas qu'un simple prestataire mais bien un conseiller et un partenaire tout au long de la marche à suivre. Le client forme donc une équipe de Key-users, constituée d'utilisateurs fonctionnels et un BPO (Business Process Owner), qui est le référent du processus métier.
Un projet comporte plusieurs phases :

- Phases de recueil du besoin :
  - Etude générale : Les consultants fonctionnels examinent avec les Key-users la situation de l'entreprise et les différents besoins. Au terme de cette étape, le cadre du projet a été défini.
  - **Conception détaillée :** Les éléments soulevés lors de l'étude générale sont approfondis et analysés.

- Phase effectuée dans l'environnement de Développement1 :
  - **Réalisation :** Lors de cette étape, les points de paramétrages nécessaires sont réalisés, ainsi que la rédaction des différentes spécifications (RICEFW) nécessaires au projet. « RICEFW » est constitué des premières lettres de Report, Interface, Conversion, Enhancement, Form et Workflow. Il s'agit de tous les points qui peuvent être soumis à des développements spécifiques afin de satisfaire le client.

- Phase effectuée dans l'environnement de Qualité2
  - **Recette et préparation :** Les Key-users testent les cas les plus délicats, les anomalies sont relevées et ajustées. Le plan de bascule vers l'environnement de Production est également préparé lors de cette phase, ainsi que la formation des utilisateurs finaux par les Key-users.

- Phase effectuée dans l'environnement de Production3
  - **Démarrage**

[1] Environnement SAP dans lequel se trouvent très peu de données. Cet environnement sert de « bac à sable » pour les tests unitaires.

[2] Environnement SAP qui est une copie de l'environnement de production. Ce n'est qu'une fois que les modifications apportées sont testées et validées qu'elles sont transportées par des « ordres de transport » dans l'environnement de production.

[3] Environnement SAP utilisé par les opérationnels. Il s'agit donc de l'environnement utilisé pour la gestion quotidienne de l'entreprise.

B) Outil utilisé

Durant mon stage j'ai essentiellement utilisé le progiciel SAP.

1. SAP

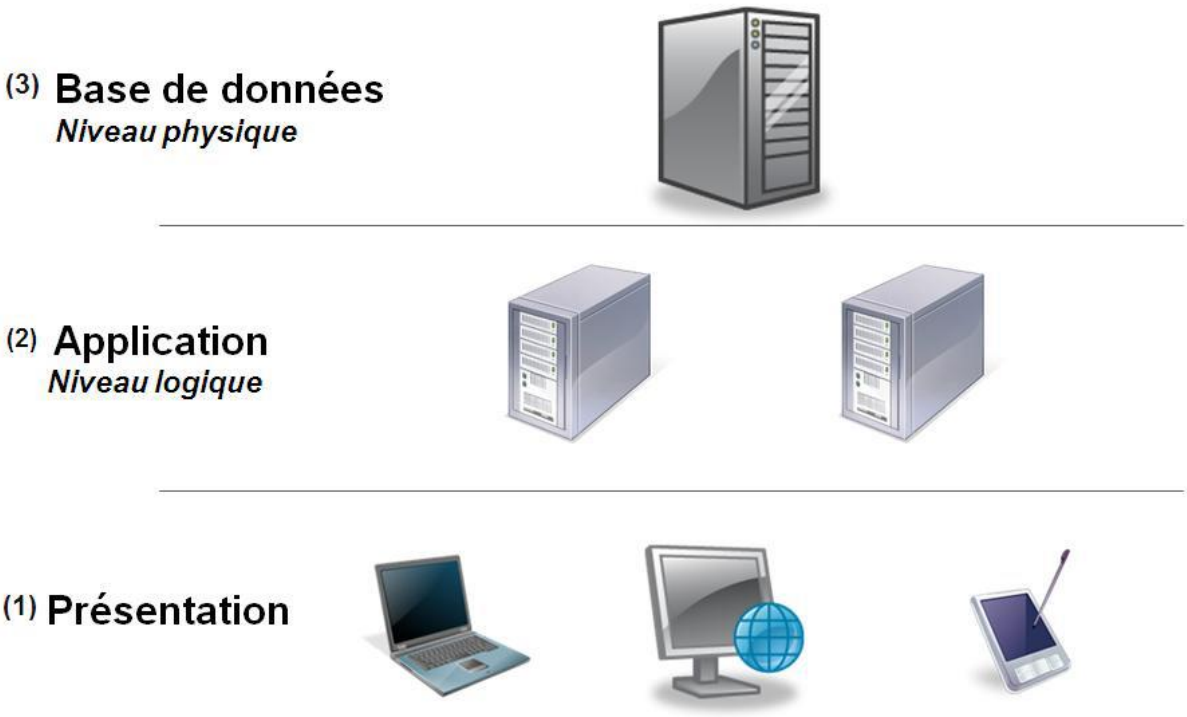
A la base SAP (Systems, applications, and Product for data processing) est une entreprise allemande qui conçoit et vend des logiciels, notamment des systèmes de gestion et de maintenance, pour des entreprises et des institutions dans le monde entier. C'est le plus gros concepteur de logiciels en Europe et le quatrième dans le monde.
Cependant par raccourci on associe souvent SAP à son célèbre SAP ERP, leader mondial dans ce domaine.

a) Le progiciel SAP *(2)*

SAP R/3 est un Progiciel de gestion intégré (en anglais ERP : Enterprise Resource Planning) destiné à optimiser les processus de gestion d'une entreprise. Il est composé d'applications standards et couvre trois grands domaines : la Finance, la Logistique et la Gestion du Personnel. Pour chacun d'entre eux, R/3 offre des fonctionnalités complètes qui permettent à l'entreprise de reproduire l'ensemble de ses flux. R/3 fonctionne avec une architecture client-serveur contrairement à R/2 qui fonctionnait sur mainframe.
Le système R/3 bénéficie d'une technologie parmi les plus avancées. Conçu de manière globale, il permet une mise en œuvre modulaire et progressive. Sa souplesse l'amène à s'adapter aux besoins spécifiques de chaque entreprise.

b) Caractéristiques de SAP R/3 *(2)*

SAP R/3 est multifonctionnel. Ses modules couvrent l'ensemble des fonctions de gestion de l'entreprise et chaque module couvre des besoins complets de gestion.
SAP R/3 est flexible. On peut installer tous les modules fonctionnels ou seulement quelques-uns. Il est entièrement paramétrable et s'adapte ainsi aux besoins et à la structure de l'entreprise. Grâce à son environnement de développement, SAP R/3 peut être adapté à des besoins spécifiques.
SAP R/3 possède une architecture client-serveur à 3 niveaux :

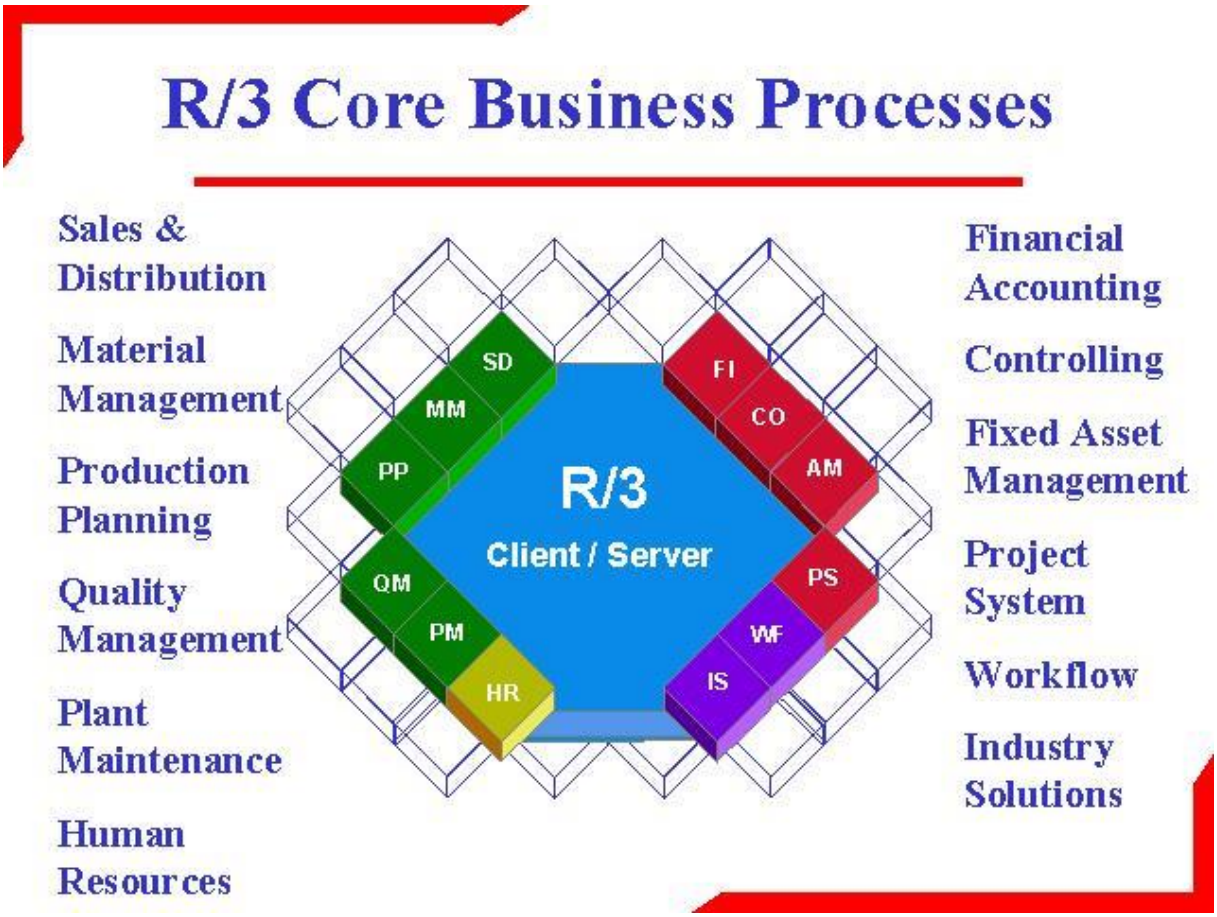


Le 1er niveau (1) se compose de serveurs de présentation (interfaces frontales) qui servent à la saisie et à l'affichage des données de l'utilisateur. Le 2ème niveau (2) est formé par les serveurs d'applications sur lesquels les traitements réels sont exécutés. Enfin au dernier niveau (3), on trouve le serveur de bases de données qui est responsable du stockage et de la mise à disposition des données.

SAP R/3 se positionne principalement sur des systèmes ouverts. Ce Progiciel est portable dans différentes bases de données (ORACLE pour AIRBUS) et possède une structure de tables qui lui est propre. Il repose sur un langage de quatrième génération (ABAP/4) adapté pour mettre en œuvre des développements spécifiques et sur son propre langage d'interrogation de base de données.

c) Les modules SAP R/3

Les modules sont les composants fonctionnels du système SAP R/3. On peut distinguer 12 modules organisés en 3 domaines fonctionnels :



_Logistique :

o Module MM (Material et Management) : gestion des achats, des stocks, des approvisionnements et des entrepôts.

o Module PP (Production et Planning) : gestion de la production et de sa planification, PIC, PDP, MRP…

o Module SD (Sales and Distribution): concerne l'administration des ventes, la gestion des prospects, des commandes clients, des expéditions et de la facturation.

o Module QM (Quality Management : gestion de la qualité) : système d'assurance qualité pour toute la chaîne logistique.

o Module PM (Planned Maintenance) : maintenance des équipements de production et des équipements généraux.

_Finance :

o Module FI (Finance : gestion financière) : système financier intégrant la comptabilité générale, client, fournisseur, la trésorerie et la consolidation du groupe.

o Module CO (Controlling : comptabilité analytique et budgétaire) : gestion par centre des opérations, compte de résultat analytique.
o Module PS (Project Systems : gestion des projets) : système intégré de gestion financière et opérationnelle de projet (planning, tâches, charges, ressources,…).

o Module TR : Trésorerie/gestion financière

o Module IM : Gestion des Investissements financiers.

_Ressources humaines :

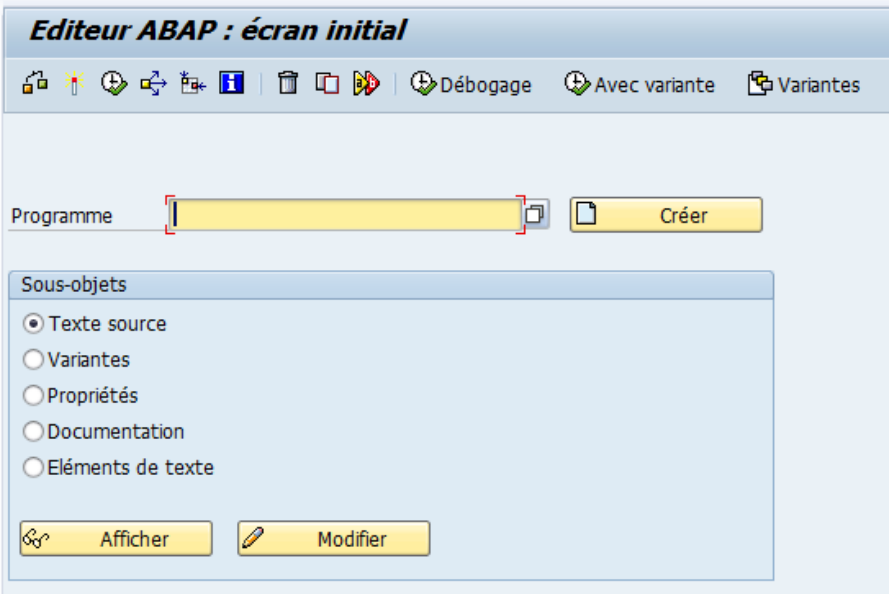
o Module HR (Human Resources : ressources humaines) : administration du personnel, gestion des temps et des frais de déplacement, planification des ressources, gestion de l'organisation et des postes, gestion de la formation et des coûts des postes.

d) Le langage ABAP/4

ABAP est un langage de programmation propriétaire, faisant partie du logiciel SAP. L'acronyme ABAP signifiant à l'origine Allgemeiner Berichts-Aufbereitungs-Prozessor (processeur générique pour la préparation de rapport) a par la suite été traduit par Advanced Business Application Programming.
À l'origine, ce langage comprenait le concept de base de données logique qui devait permettre à tout utilisateur de créer lui-même les rapports dont il avait besoin. Il s'est avéré en fait que développer des programmes avec ce langage ne pouvait être fait en pratique que par des professionnels disposant d'une solide expérience.

e) Navigation dans SAP

La navigation dans SAP se fait via des transactions. Ces dernières sont un identifiant à 4 caractères avec généralement 2 lettres puis 2 chiffres, par exemple SE37. Elles permettent la succession d'écrans appelées Dynpros, chaînés entre eux par un programme contenant la logique de la transaction.
Ces transactions sont la base de l'utilisation de SAP, tous les traitements peuvent être exécutés via des transactions. Dans l'entête de tout écran SAP, il y a une zone de saisie permettant d'entrer le nom d'une transaction et de l'exécuter.



Par exemple, la transaction SE38 permet de faire des développements ABAP, la transaction SE11 permet de consulter les tables ou encore la transaction CO03 permet de consulter un ordre de fabrication. Il existe des centaines de transactions disponibles sous SAP.

On peut aussi atteindre un programme SAP grâce à l'arborescence des transactions organisées suivant les différents modules (logistique, finance, RH,…). Mais cette technique est beaucoup plus longue que celle de renseigner directement le code de la transaction.

f) Informations sur le code ABAP

Le code source est écrit directement dans un composant de SAP appelé SAP Workbench, il est stocké dans la base d'information de l'outil. Ce langage a besoin d'être compilé pour pouvoir être exécuté.

À chaque modification d'un programme, celui-ci doit être transporté de l'environnement dans lequel il est créé vers un environnement de test puis dans l'environnement de production (il peut exister plus de trois environnements). Cette opération est réalisée par la mise en œuvre d'un ordre de transport. Ceci est une question de sécurité pour ne pas faire passer un programme déficient en production et ainsi éviter de gros risques.

g) Exemples de réalisations avec le code ABAP *(3)*

Il peut être utilisé pour réaliser :

- Des rapports : ABAP a été conçu à l'origine pour cela. Un rapport est constitué en général d'un écran de sélection de données et d'une liste affichant le résultant. Les rapports peuvent aussi être générés par un utilisateur avec SAP Business Report.

- Des transactions : une transaction est constituée d'une succession d'écrans appelés Dynpro, chaînés entre eux par un programme contenant la logique de la transaction.

- Des fonctions : il s'agit de sous-programmes contenant la description d'une fonctionnalité du système. Ces fonctions peuvent être internes et permettre ainsi une plus grande cohérence du système ou permettre la communication avec d'autres systèmes. Les appels peuvent être réalisés à partir d'autres langages, on parle alors de BAPI.

- Des programmes de gestion d'IDOCs. Un Idoc est un document de format unique composé de segments qui, une fois rassemblés et exécutés permettent diverses opérations au sein du système (création de commandes client, de demandes d'achats, gestion de données du personnel…). L'avantage d'un Idoc est, que de par sa composition, il peut provenir de systèmes externes pour être reconstruit dans SAP et inversement.

ABAP a évolué pour intégrer la programmation objets. Tous les concepts de la programmation objets se retrouvent dans ABAP Objects. ABAP Objects constitue une extension du langage ABAP, avec lequel il est totalement compatible : ainsi il est possible qu'un programme comporte à la fois une partie procédurale et une partie orientée objets, car le déroulement procédural du programme peut à tout moment instancier une classe puis appeler les méthodes de l'objet qu'il vient de créer.

II) Travaux réalisés

Mon premier mois de stage a été une période de formation en interne qui a consisté à découvrir le logiciel SAP, son langage l'ABAP grâce à de la documentation ainsi que des exercices d'applications.
Suite à cela j'ai été placé sur un projet afin de finir cette formation.

A) Création de BAPI pour l'entreprise ALTERBURO

1. Présentation du contexte

ALTERBURO est une société spécialisée dans la distribution de fournitures de bureau, mobilier et consommables informatiques.

Augusta Reeves s'est engagé à créer pour cette entreprise un outil pour gérer les différents flux générés par leur activité. L'interface devant être réalisée en JAVA (résultat beaucoup plus esthétique que ce qui est réalisable avec ABAP) il fallait créer des BAPI pour faire l'intermédiaire entre la BDD et l'interface.
Comme énoncé précédemment une BAPI est une fonction pouvant être appelée depuis un autre système. Dans ce cas il est donc possible de saisir des paramètres dans l'interface JAVA, exécuter la BAPI avec ces paramètres dans le système SAP puis renvoyer les paramètres de sortie dans l'interface.

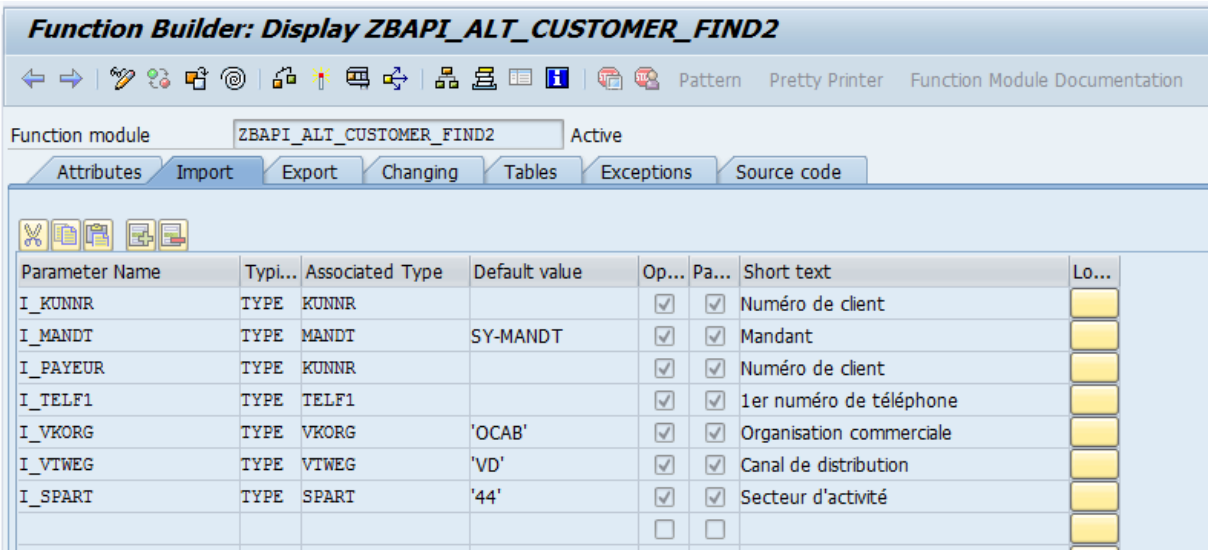
Durant cette période j'ai travaillé en relations avec 3 consultants techniques juniors et 2 consultants expérimentés (l'un fonctionnel et l'autre technique).
Le rôle du consultant fonctionnel est d'échanger avec le client pour déterminer son besoin (chaque fonctionnel est spécialisé dans un voire deux modules SAP. Il rédige ensuite une spécification fonctionnelle (documentation détaillant le besoin) destinée au consultant technique (qui lui n'est pas spécialisé dans un module) qui créera le programme en fonction de celle-ci.

Les travaux qui m'ont été demander de faire durant cette période n'étaient pas très compliqués et un peu répétitif, ils m'ont permit de bien me familiariser avec les bases du langage et de développer des automatismes de codage. Ils m'ont également permis de connaître petit à petit les principales tables et variables des modules sur lesquels j'ai travaillé.

2. Création des BAPI

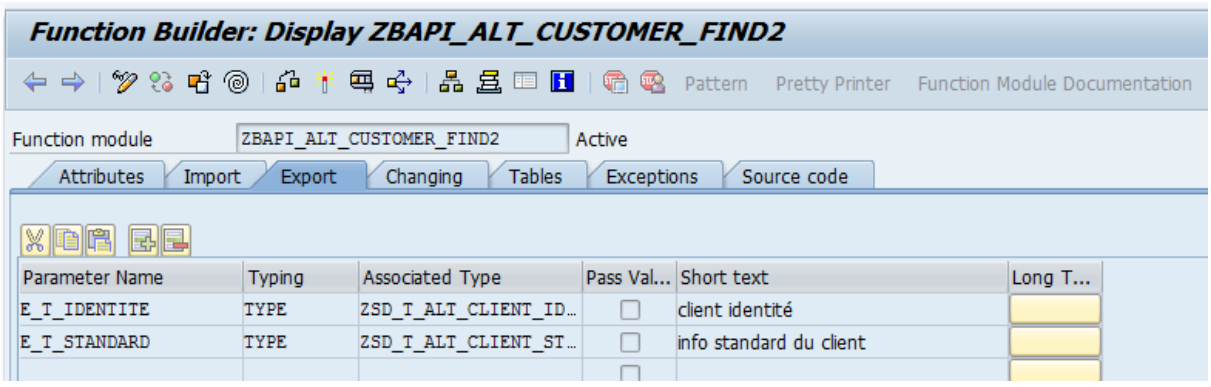
Lors de la création d'une BAPI il faut dans un premier temps spécifier les paramètres (d'entrée et de sortie).
Pour chaque paramètre d'entrée il faut spécifier :

- son nom (p_variable)
- son type (il est possible de leur donner un type élémentaire que l'on spécifie (type) ou directement le type d'une variable déjà existante (like))
- la variable à laquelle on veut faire référence
- s'il est obligatoire ou optionnel
- sa désignation



Parameter Name	Typi...	Associated Type	Default value	Op...	Pa...	Short text	Lo...
I_KUNNR	TYPE	KUNNR		☑	☑	Numéro de client	
I_MANDT	TYPE	MANDT	SY-MANDT	☑	☑	Mandant	
I_PAYEUR	TYPE	KUNNR		☑	☑	Numéro de client	
I_TELF1	TYPE	TELF1		☑	☑	1er numéro de téléphone	
I_VKORG	TYPE	VKORG	'OCAB'	☑	☑	Organisation commerciale	
I_VTWEG	TYPE	VTWEG	'VD'	☑	☑	Canal de distribution	
I_SPART	TYPE	SPART	'44'	☑	☑	Secteur d'activité	
				☐	☐		

Pour les paramètres de sorties les champs à spécifiés sont les mêmes excepté la faculté de la variable.



Parameter Name	Typing	Associated Type	Pass Val...	Short text	Long T...
E_T_IDENTITE	TYPE	ZSD_T_ALT_CLIENT_ID...	☐	client identité	
E_T_STANDARD	TYPE	ZSD_T_ALT_CLIENT_ST...	☐	info standard du client	
			☐		

Enfin dans l'onglet Source se trouve le code à proprement parler, c'est dans cette partie qu'est effectué le traitement en fonction des paramètres d'entrées.
Dans cette partie il est bien entendu possible de faire appel à d'autres BAPI.

3. Taches réalisées

J'ai travaillé sur des développements essentiellement en rapport avec les modules SD et MM (articles, commandes, ventes, livraisons, …).

Au fur et à mesure que les spécifications étaient rédigées, le consultant technique me les fournissais chacune accompagnée d'un temps de développement prévisionnel.
A chaque fin de développement je rédigeais une spécification technique indiquant les paramètres d'entrée et de sortie du programme (en précisant leur nom et leur type), la logique d'exécution du programme (au cas où il doit être repris dans le futur). Cela en plus de commenter les différentes parties du code.

Je vais maintenant détailler quelques développements que j'ai effectués.

a) Sélection des villes en fonction du code postal

Une des premières BAPI qui m'a été demandé de faire consistait à renvoyer les villes correspondant au code postal entré en paramètre.

Avant exécution :

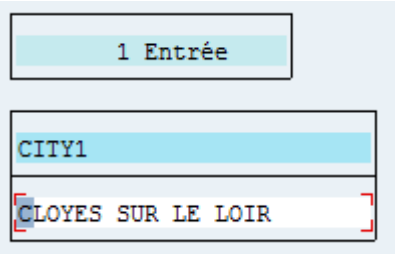
Paramètres d'import	Valeur
I_MANDT	800
I_POST_CODE1	28220

Après exécution :

Paramètres d'import	Valeur
I_MANDT	800
I_POST_CODE1	28220

Paramètres export	Valeur
E_VILLE	1 Entrée

Ouverture du tableau (ici à une entrée) :



Si aucun code postal n'était renseigné en entrée il fallait renvoyer toutes les communes contenues dans la table.
J'ai donc fondé le programme à partir de cet algorithme :

```
Si I_POST_CODE1 est vide
     SELECT CITY1
     FROM table
     INTO E_ville
Sinon
     SELECT ville
     FROM table
     INTO E_ville
     WHERE CP = I_POST_CODE1
Fin si
```

J'ai réalisé un grand nombre de BAPIs de ce genre pour diverses données.

b) Sélection des interlocuteurs de niveau supérieur à celui renseigné

Voici une BAPI plus complexe que j'ai eu à réaliser pendant ce projet.
Le but de cette fonction est d'obtenir la liste des interlocuteurs pour un client donné (leur numéro, nom et prénom contenus dans les tables gérant les clients) de niveau supérieur à celui renseigné en paramètre d'entrée (l'interlocuteur est nommé en dessous PARNR).
Si aucun interlocuteur n'est renseigné en paramètre on renvoi tous les interlocuteurs.

Les différents niveaux sont :

- K : Sans pouvoir
- W : PouvoirRéglementaire
- P : Procuration
- M : Codécideur
- A : Décideur
- H : Décideur principal
- G : Pouvoir général

Cependant rien dans SAP ne permet de les classer par ordre d'importance.
J'ai donc créé un rang dans le programme en classant les différents niveaux.

KUNNR est le client (paramètre d'import obligatoire)
PARNR est le code interlocuteur (paramètre d'import facultatif)
PARVO est le niveau de l'interlocuteur
I_XXX signifie que XXX est une variable importée
R_PARVO contient les valeurs de PARVO à retourner en fin de programme

La logique du programme est la suivante :

Sélection du niveau d'interlocuteur selon le code partenaire et le client

```
Si I_PARNR est renseigné.
     SELECT PARVO
     FROM table
     INTO PARVO
     WHERE PARNR = I_PARNR
     AND KUNNR = I_KUNNR

     Si PARVO n'est pas vide
          Valeur de PARVO
               Quand 'K'
                    On remplit R_PARVO avec tous les PARVO de W à G
               Quand 'W'
                    On remplit R_PARVO avec tous les PARVO de P à G
               Etc…
                    Etc…
               Quand 'H'
                    On remplit R_PARVO avec tous le PARVO G
          Fin Valeur
     Sinon
          On remplit R_PARVO avec tous les PARVO de K à G
     Fin Si
Sinon
     On remplit R_PARVO avec tous les PARVO de K à G
Fin Si
```

Création de la liste des niveaux interlocuteur à récupérer

```
Si R_PARVO n'est pas vide
     SELECT PARNR NOM PRENOM PARVO
     FROM table
     INTO table_interne
     WHERE PARVO IN R_PARVO
     AND KUNNR = I_KUNNR
```

Récupération des interlocuteurs selon le niveau renseigné en entrée

```
SELECT PARVO DESIGNATION
FROM table_désignation
INTO table_désignation_interne
POUR TOUTES LES ENTREES DANS table_interne
WHERE PARVO = table_interne-PARVO
```

Récupération des désignations des niveaux d'interlocuteur (contenus dans une autre table)

```
Boucle sur table_interne
LECTURE DE table_désignation_interne avec comme clé table_interne-PARVO
table_interne-désignation = table_désignation_interne-désignation
Fin Boucle
```

E_INTERLOCUTEUR[] = table_interne[]

Fin Si

Ajout des désignations de niveaux dans la table interne contenant les interlocuteurs de niveau supérieur.

Transfert du contenu de la table interne dans le contenu de la table à exporter.

Voici un exemple d'utilisation pour le client A4000000 sans partenaire précisé :

Paramètres d'import	Valeur
I_MANDT	800
I_KUNNR	A4000000
I_PARNR	0000000000

Après exécution :

Paramètres d'import	Valeur
I_MANDT	800
I_KUNNR	A4000000
I_PARNR	0000000000

Paramètres export	Valeur
E_INTERLOCUTEUR	9 Entrées

Ouverture du tableau (à 9 entrées), on retourne tous les partenaires de ce client :

9 Entrées

PARNR	NAMEV	NAME1	P	VTEXT
0000059972	ELSIABETHe	TONNER		
0000060011	PATRICKE	PERNOTE	H	Décideur principal
0000060012	PATRICKE	PERNOTE	H	Décideur principal
0000060076	MARLENE	SASSEUR	G	Pouvoir général
0000060077	JEAN	NEYMAR	W	PouvoirRéglementaire
0000060078	PAUL	HOCHON	P	Procuration
0000060079	YVES	REMAURE	M	Codécideur
0000060081	JACQUES	SELERE	A	Décideur
0000060082	JEREMY	MACULOTTE	K	Sans pouvoir

Voici maintenant un exemple avec un partenaire renseigné, le programme nous retournera uniquement les partenaires qui sont de niveau supérieur au sien.

Client A4000000 pour précédemment, partenaire 60079 qui est de niveau P :

Paramètres d'import	Valeur
I_MANDT	800
I_KUNNR	A4000000
I_PARNR	0000060079

Après exécution :

Paramètres d'import	Valeur
I_MANDT	800
I_KUNNR	A4000000
I_PARNR	0000060079

Paramètres export	Valeur
E_INTERLOCUTEUR	4 Entrées

Ouverture du tableau (à 4 entrées), on retourne tous les partenaires de ce client dont le niveau est supérieur au niveau P :

4 Entrées

PARNR	NAMEV	NAME1	P	VTEXT
0000060011	PATRICKE	PERNOTE	H	Décideur principal
0000060012	PATRICKE	PERNOTE	H	Décideur principal
0000060076	MARLENE	SASSEUR	G	Pouvoir général
0000060081	JACQUES	SELERE	A	Décideur

B) Amélioration du système déjà mis en place chez Flammarion suite à son évolution

1. Présentation du contexte

Madrigall est une holding regroupant deux structures : Gallimard et Flammarion, Gallimard ayant racheté Flammarion en 2009. Ces deux structures sont deux prestigieuses maisons d'édition françaises. Chacune regroupe plusieurs maisons d'édition : autour de Gallimard se trouvent les éditions Gallimard Jeunesse, Denoël, Table Ronde, etc. et le groupe Flammarion possède, quant à lui, entres autres, les éditions Casterman, J'ai Lu, Autrement, ou encore Audie. Chacune a également une entité de diffusion, qui correspond à la partie commerciale : CDE (Centre de diffusion de l'édition) pour Gallimard et Flammarion Diffusion pour Flammarion, ainsi qu'une entité de distribution, qui produit les livres et les acheminent jusqu'aux libraires : SODIS pour Gallimard et UD (Union Distribution) pour Flammarion.

Gallimard a actuellement le logiciel comptable Qualiac ainsi que différents autres logiciels interfacés avec Qualiac. Flammarion a, quant à lui, SAP depuis 2004. SAP avait été implanté par Accenture lorsque le groupe appartenait à un groupe italien.

Les deux groupes, centralisés par la nouvelle holding, désirent harmoniser petit à petit leurs outils de gestion afin de faciliter la cohésion au sein de la nouvelle entité et de pouvoir éditer un *report*ing de groupe. La décision a donc été prise d'implanter SAP dans une des sociétés de Gallimard, la SODIS, qui s'occupe de la distribution du groupe. Le projet consistait donc à mettre en place les modules SD, FI et CO à la SODIS, ainsi qu'à remanier le système déjà présent chez Flammarion pour le rendre plus adapté aux besoins actuels.

La mission sur laquelle j'ai été affecté sur le projet Madrigall est la réalisation de développements afin de répondre aux besoins de la SODIS et aux évolutions souhaitées par Flammarion.

J'ai commencé à travailler sur ce projet dès le début du mois de Juillet. Toute l'équipe travaillait directement dans les locaux du client. Cela a un avantage car on était entre personnes travaillant sur le même projet et nous pouvions avoir les réponses à nos questions beaucoup plus rapidement (que ce soit des problématiques à régler avec des consultants fonctionnels ou avec le client). Cela a permis de faciliter les échanges et donc simplifier le travail.
Lorsque nous sommes arrivé, l'équipe technique, la plupart des spécifications fonctionnelles étaient rédigées ainsi nous avons pu commencer à rentrer dans le vif du sujet directement.

Nous avons eu beaucoup de travail car Flammarion travaille très peu avec les programmes standards de SAP, qui ne lui conviennent pas forcément. Il faut donc créer des programmes appelés spécifiques.

2. Taches réalisées

Dans un premier temps j'ai lu des spécifications dans le but de m'imprégner du sujet et d'identifier des points qui n'étaient pas suffisamment expliqués et pouvaient donc aboutir à des développements ne correspondant pas au besoin du client.
J'ai beaucoup échangé avec les consultants fonctionnels et le client pour bien assimiler le besoin mais aussi le fonctionnement des flux de l'entreprise. Chose essentielle pour mener à bien les développements.

Dans un second temps j'ai commencé à réaliser des développements, toujours en consultant le client ou les fonctionnels lorsque j'en avais besoin.

La première partie de mes programmes a porté sur les offices.

a) L'office

L'office (vocabulaire spécifique au monde de l'édition) est un mode d'approvisionnement des librairies qui concerne les nouveautés. C'est un contrat par lequel le libraire s'engage auprès d'un fournisseur à lui commander un certain volume de livres parmi les nouveautés et qui lui permet de renvoyer les invendus, plus de trois mois et moins de douze mois après la parution (en ne payant que les frais de ports pour les livres invendus).
La grille d'office (rempli par le libraire) sert à définir un office par défaut. Elle indique la quantité souhaitée pour chaque catégorie de livres (classés par thème et par importance de tirage).

Suspension des grilles d'office

Sur cette partie, le premier programme sur lequel j'ai eu à travailler a été la suspension temporaire des grilles pour un ou plusieurs clients
Le besoin était de pouvoir saisir une suspension temporaire au niveau de la grille (Saisie rapide) et non pas au niveau du groupe (qui dépend de la grille).

Voici le format du tableau qu'il fallait gérer avant :

Client	Code grille	Lib grille	Groupe	Lib groupe	Quantité	Date début susp	Date fin susp
Librairie 1	ASE	Actes Sud Editions	1	Grandes ventes	1	01/07/2014	31/08/2014
Librairie 1	ASE	Actes Sud Editions	3	Junior	1	01/07/2014	31/08/2014
Librairie 1	ASE	Actes Sud Editions	16	Poches	0	01/07/2014	31/08/2014
Librairie 1	FLM	Flammarion	1	Grand tirages	1		
Librairie 2	FLM	Flammarion	2	Moyens tirages	2		
Librairie 2	FLM	Flammarion	3	Petits tirages	4		
Librairie 2	HOE	Editions Hoebeke	3	BD	3		

Ces champs sont ceux contenus dans la table ZSDGRI_CLI.

Il fallait inscrire manuellement les dates pour chaque ligne devant être suspendu sachant que les suspensions se font par client et grille.
On pouvait également se servir de cette transaction (ZSDGRCLI) pour rajouter des lignes.
L'idée a donc été de séparer ces deux actions : l'ajout de ligne et les saisies de dates.

Il m'a fallu créer premièrement une autre transaction (ZSD_CLI_SUP_OFF) pour gérer les dates avec comme liste de champs le client, le code grille, son libellé, les dates de début et fin et enfin supprimer de l'écran actuel l'affichage des dates.
La suppression des dates dans la transaction existante a été simple, il a fallu les supprimer de l'écran dans lesquels ils étaient affichés.

La création de la transaction ZSD_CLI_SUP_OFF m'a demandé plus de temps, j'ai dû construire le programme en partant pratiquement à zéro.
J'ai commencé par sélectionner les champs dont j'allais avoir besoin.
Le résultat pour une sélection égale à celle réalisée précédemment donne ce résultat :

Client	Code grille	Lib grille	Date début susp	Date fin susp
Librairie 1	ASE	Actes Sud Editions	01/07/2014	31/08/2014
Librairie 1	FLM	Flammarion		
Librairie 2	FLM	Flammarion		
Librairie 2	HOE	Editions Hoebeke		

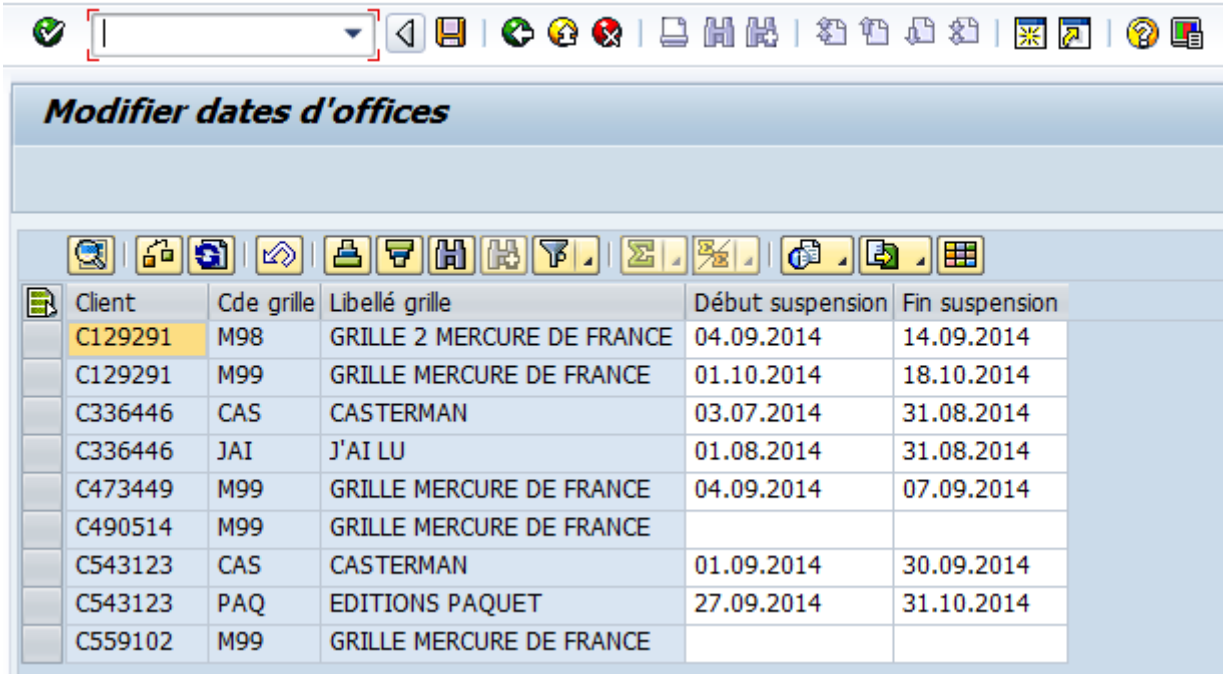
Nous pouvons voir que les doublons disparaissent, la saisie des dates se fait une seule fois par client et grille ce qui permet de gagner du temps.
J'ai indiqué les champs dates comme éditables et les autres champs comme non éditables.
J'ai ensuite rajouté la fonctionnalité de sauvegarde.
Pour cela je récupère les données du tableau que je stock dans une table interne.
Je n'enregistre que les lignes ayant été modifiées pour maximiser la rapidité d'exécution du programme.

T1 = Table des données d'origine
T2 = Table des données lors de l'enregistrement

Je me sers de l'algorithme suivant pour identifier les lignes retouchées :

```
Boucle sur T2
     Lecture T1 avec client et grille en guise de clé.
     Si T1-date_début <> T2-date_début OU T1-date_fin <> T2-date_fin
          Mise à jour de la table de BDD avec les nouvelles valeurs
          Quand client = T1-client ET grille = T1-grille
     Fin si
Fin boucle
```

Voici l'écran final de la transaction que j'ai créée.



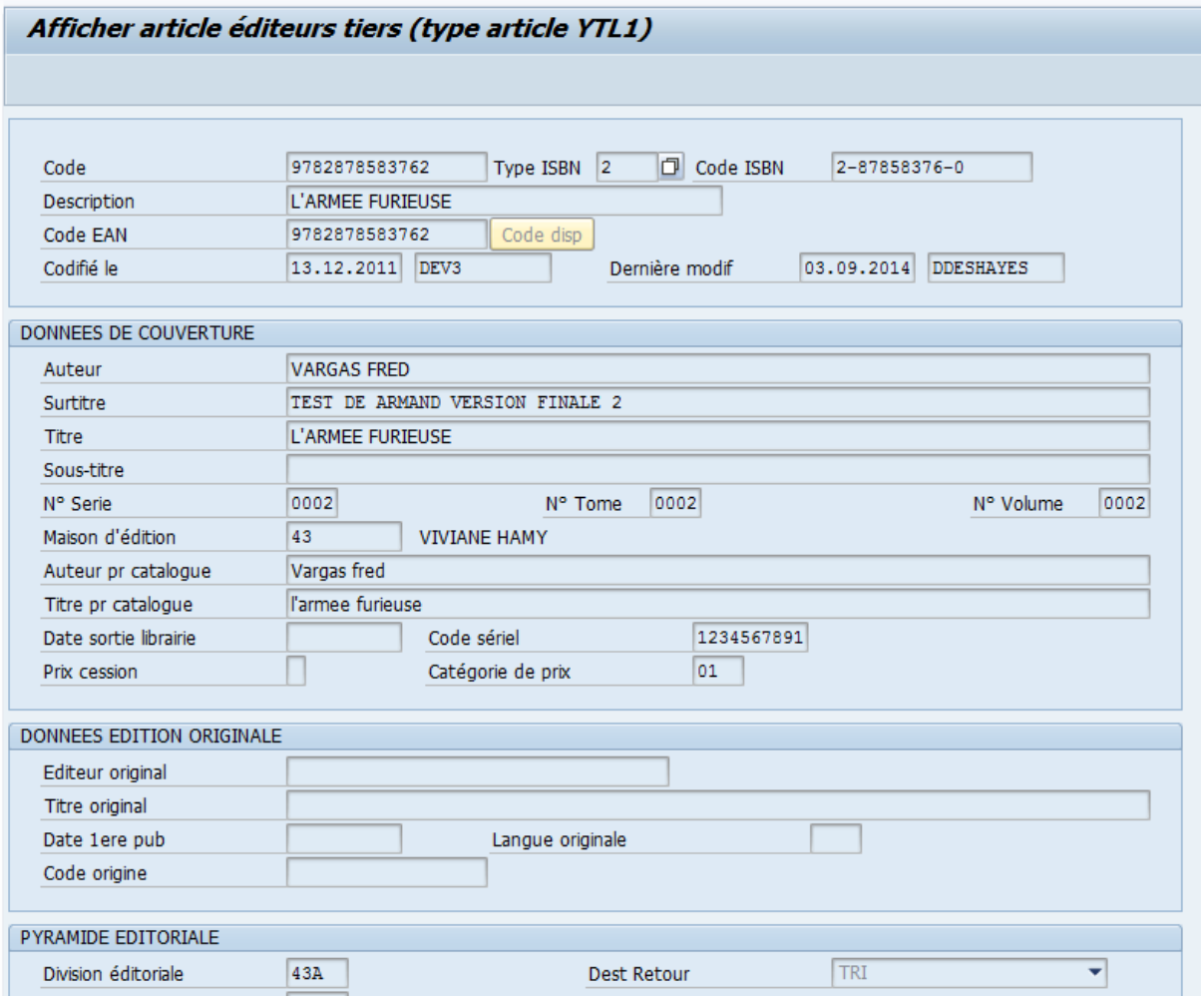
Client	Cde grille	Libellé grille	Début suspension	Fin suspension
C129291	M98	GRILLE 2 MERCURE DE FRANCE	04.09.2014	14.09.2014
C129291	M99	GRILLE MERCURE DE FRANCE	01.10.2014	18.10.2014
C336446	CAS	CASTERMAN	03.07.2014	31.08.2014
C336446	JAI	J'AI LU	01.08.2014	31.08.2014
C473449	M99	GRILLE MERCURE DE FRANCE	04.09.2014	07.09.2014
C490514	M99	GRILLE MERCURE DE FRANCE		
C543123	CAS	CASTERMAN	01.09.2014	30.09.2014
C543123	PAQ	EDITIONS PAQUET	27.09.2014	31.10.2014
C559102	M99	GRILLE MERCURE DE FRANCE		

b) La fiche article

La fiche article est un point important, en effet c'est la transaction qui permet de créer les articles donc la base du fonctionnement de l'entreprise. Beaucoup de développement en dépendent directement. Cette transaction sert à créer, afficher ou modifier les articles.

Ajout de champs dans la fiche articles

La fiche article correspond à une transaction servant à créer, modifier ou afficher un article.
Il existe une transaction standard pour cela mais Flammarion utilise une transaction spécifique ne reprenant que les champs les plus fréquemment utilisé (par gain de temps et par praticité).



Sur la fiche article j'ai réalisé des développements plutôt simples (ajouts de champs sur l'interface).
En effet certains champs qui n'étaient pas utilisés dans le passé doivent l'être aujourd'hui.

Pour ce faire je déclarais dans un premier temps la nouvelle variable dans le programme.

```
* AFR 24/07/2014 Pyramide Edit correction
  dsc_zsurcoll TYPE char40, "ZTSD_SURCOLL-libelle
  dsc_zgredit TYPE char30, "ZTSD_GREDIT-libelle
  dsc_zdiffart TYPE char30, "ZTSD_DIFFART-libelle
* End AFR
* AFR 06/089/2014 Fiche article correction E6
      prat9                LIKE mvke-prat9,
      prat10               LIKE mvke-prata.
* End AFR
```

La seconde partie du travail contenait plusieurs possibilités, en effet il a fallait gérer les 3 modes possibles de connexion à la transaction ce qui fait deux cas à traiter : la sélection des données et le affichage, et leur enregistrement. Le mode modification étant un mélange des deux.

Pour afficher les donnée dépendant de l'article j'ai réalisé un select avec comme condition le numéro de l'article. Pour les données de types libellés (qui sont dans des tables contenant uniquement les descriptions des codes utilisés) j'ai réalisé un select en fonction du code et j'ai alimenté le champ avec la valeur récupérée.

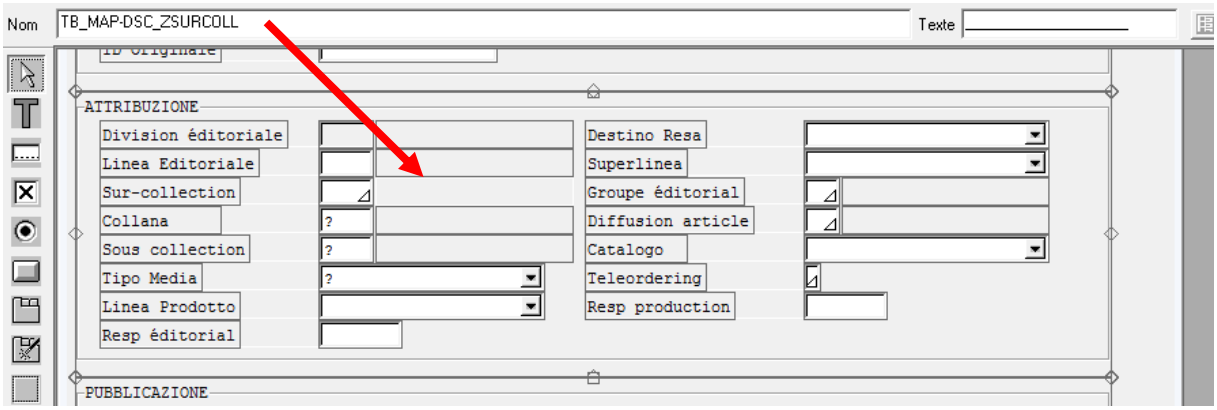
```
* AFR 24/07/2014 Pyramide Edit correction
  SELECT SINGLE libelle
       FROM ztsd_surcoll
       INTO tb_map-dsc_zsurcoll
       WHERE code = zamtcoll-zsurcoll.

  SELECT SINGLE libelle
    FROM ztsd_gredit
    INTO tb_map-dsc_zgredit
    WHERE code = zsdtedit-zgredit.

  SELECT SINGLE libelle
      FROM ztsd_diffart
      INTO tb_map-dsc_zdiffart
      WHERE code = zamtcoll-zdiffart.
* End AFR
```

- Sélection du libellé
- Depuis la table contenant les libellés des sur-collections
- Dans le champ prévu à cet effet
- Quand le code est égal au code sur-collection de l'article affiché

Enfin j'ajoutais la variable dans l'écran Dynpro et son libellé.
Voici par exemple l'ajout du libellé de la sur-collection. Le texte est renseigné en fonction de la valeur contenue dans le champ tb_map-des_zsurcoll.
Nous pouvons voir que les zones contenant les libellés sont grisées. En effet il n'est pas possible de modifier les libellés depuis la fiche article.



Dans le cas d'enregistrement de champ j'insérais la valeur récupérée dans la variable et je l'enregistrais dans la table associée en fonction du numéro d'article.
Les libellés ne sont pas enregistrés, car ils ne sont pas éditables. On gère les attributs uniquement par leur code. Si une modification de libellé doit être apportée il existe un programme dédié à cela.

Voici un exemple d'enregistrement de données.
Je récupère les valeurs des champs prat9 et prat10, et les déplace respectivement dans prod_att_9 et prod_att10. Ils sont maintenant dans une structure nommée salesdata qui va être passé en paramètre d'une BAPI standard qui gère l'enregistrement dans les tables.

```
MOVE tb_map-prat9 TO salesdata-prod_att_9. "AFR fiche articles E6 04/08/14
MOVE tb_map-prat10 TO salesdata-prod_att10. "AFR fiche articles E6 04/08/14
```

c) La pyramide éditoriale

La pyramide éditoriale est composée de plusieurs niveaux, chaque niveau héritant des attributs des niveaux précédent :

- La maison d'édition
- La division éditoriale
- La ligne éditoriale
- La sur-collection (champs rajouté pendant le projet)
- La collection
- La sous-collection

Le livre arrivant après, juste sous la sous-collection.

Initialisation de données articles à partir de la collection

Lors de la création d'un article (par la fiche article) beaucoup de champs sont renseignés automatiquement lorsqu'ils sont des attributs de la pyramide éditoriale.
Le besoin de la SODIS était de rajouter des champs dans la fiche article (dont certains dépendent de la pyramide éditoriale).
Ils ont donc demandé à ce que ces champs soient initialisés spontanément à la création de l'article pour éviter aux utilisateurs de le faire dans le futur.

Ces champs sont donc non éditables en création pour ne pas induire en erreur les utilisateurs mais le sont en modification.

Pour les initialiser j'ai sélectionné leurs valeurs dans les tables associées de la pyramide éditoriale puis les ai affectés dans les champs de la fiche article après validation par l'utilisateur, juste avant l'appel de la sauvegarde.

Création d'une fiche collection

Sur ce domaine j'ai été amené à créer un programme pour gérer les collections. Ce programme n'existait pas chez Flammarion, ils créaient des collectons via une transaction peu pratique à utiliser que peu de personnes maitrisaient.

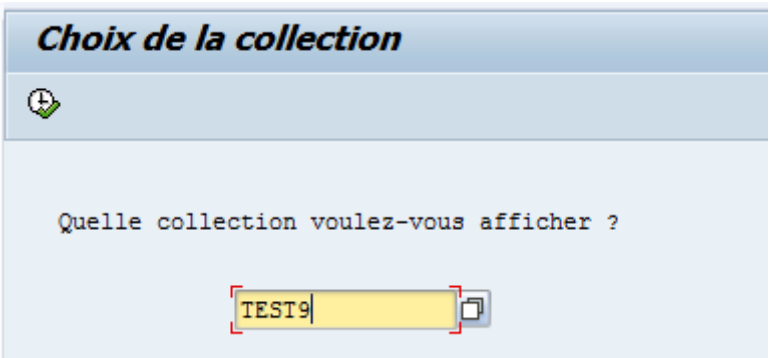
Ce développement a été celui qui m'a demandé le plus de temps.
Il est composé de quatre parties :

- La création de fiche collection
- La modification de fiche collection
- L'affichage de fiche collection
- La duplication de fiche collection

La transaction est constituée de 3 écrans.
Le premier, de lancement, est un écran de choix entre les différentes actions à réaliser.

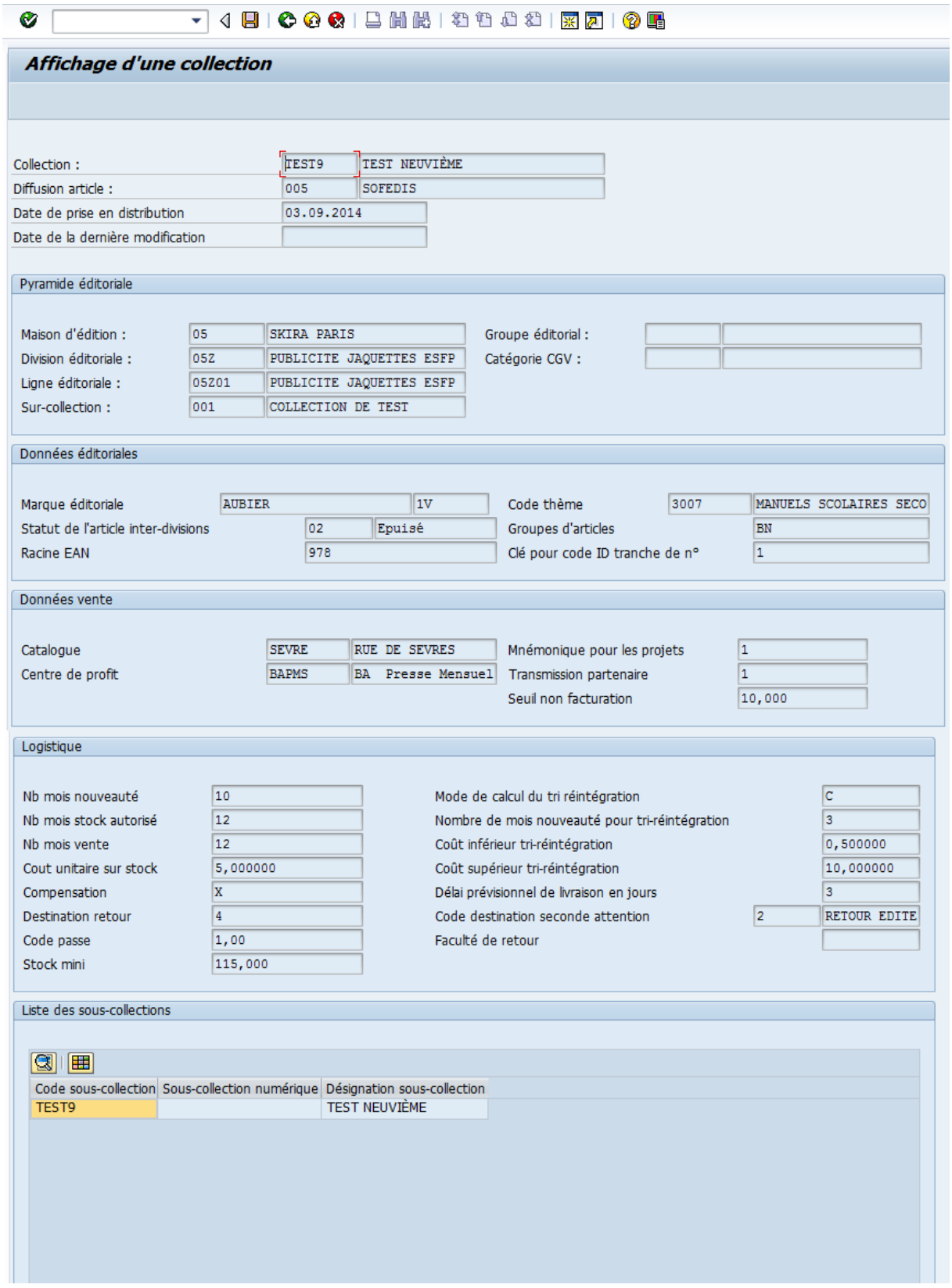
Si le choix est la création de programme, le troisième écran est appelé. Il comporte les attributs de la collection. Il est appelé vide.

Si le choix est une des 3 autres propositions le second écran est appelé, c'est un écran de sélection où il faut renseigner le code d'une collection



La question est une variable qui est renseignée en fonction du bouton cliqué dans le premier écran (modifier, afficher ou dupliquer).

Lors de la validation j'effectue plusieurs vérifications sur le numéro de collection, premièrement s'il est renseigné et ensuite s'il existe dans la base de données. Si ces deux vérifications sont bonnes j'appelle le troisième écran que j'alimente en fonction de la collection et de l'opération
Lors d'un affichage, tous les champs sont récupérés dans la BDD et grisés sur l'écran :



Lors d'une modification la collection, les données de la pyramide éditoriale, les champs qui en dépendent et bien entendu tous les libellés sont grisés.
Lors d'une duplication les données de la pyramide éditoriale, les champs qui en dépendent et les libellés sont grisés à part le libellé de la collection. La/les sous collections existante(s) sont supprimées.

J'ai créé un sous-programme qui permet la sélection des données, je l'appelle avant l'affichage, la modification ou la duplication.
Dans ce programme sont sélectionnées dans un premier temps les données qui dépendent de la collection.
Dans un second temps je sélectionne les données dépendant des champs de la pyramide éditoriale.
Enfin je récupère les libellés des codes dans chaque table (cette phase est répétitive car il faut effectuer une vingtaine de select dans des tables différente).

La préparation de l'enregistrement est différente si on est en création, en modification ou en duplication.

- En création ou duplication

Des vérifications sont effectuées sur le code collection. Je contrôle la longueur qui doit être de 5, et si le code saisit n'existe pas déjà). J'ai nommé ce sous-programme « **check_coll** ».

Si ces deux points sont corrects on passe à la vérification de la pyramide éditoriale. En effet elle doit être cohérente et déjà exister (on ne crée pas des attributs de la pyramide depuis cette transaction). Pour cela je sélectionne la ligne éditoriale (dans la table consacrée) en prenant en compte dans la requête des différents champs de la pyramide. Si la requête retourne quelque chose la pyramide est correcte, on peut donc passer à la vérification suivante. J'ai nommé ce sous-programme « **verif_pyram** ».

La troisième vérification est celle du groupe éditorial. Je vérifie s'il est renseigné et s'il correspond bien à la maison d'édition renseignée (c'est un de ses attributs). Ce sous-programme est « **verif_gredit** ».

La dernière vérification est celle de la sous-collection. La sous-collection dépend directement de la collection. Chaque collection a au minimum une sous-collection (qui peut avoir le même code et le même libellé que la collection). Une des demandes du client était la création automatique d'une sous-collection avec les mêmes valeurs que la collection au cas où celle-ci n'était pas précisée. Pour cet attribut je commence par supprimer les doublons s'il y en a car la saisie est réalisé par un utilisateur dans un tableau, il peut donc y avoir des erreurs. Ensuite je vérifie si la sous-collection n'est pas renseignée et que la sous-collection ou le libellé l'est. Enfin je contrôle

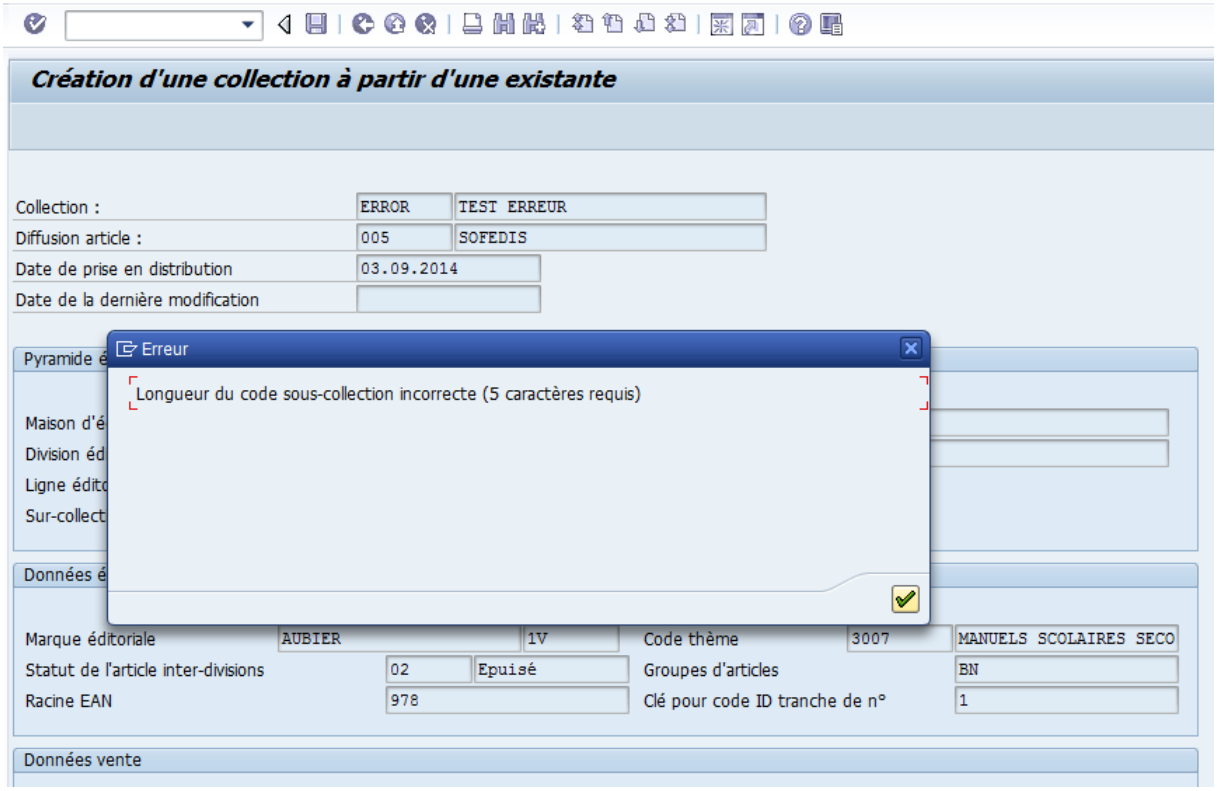
le format de la sous-collection pour voir s'il est bien sur 5 caractères. Ce sous-programme est « **verif-sscoll** ».

- En modification

verif-sscoll uniquement.

A la suite de ces vérifications j'ai créé des messages d'erreur correspondant aux différents cas de figures résultant des vérifications. J'affiche les messages sous forme de pop-up et bloque l'enregistrement, on est renvoyé sur la fiche pour corriger les erreurs. Il est donc impossible d'enregistrer des données incohérentes.

Voici un exemple de message d'erreur (j'ai renseigné une sous-collection à 3 caractères pour le test :



L'enregistrement à proprement parler est relativement le même que l'on soit en création, modification ou duplication.

J'ai créé un sous-programme qui récupère les données de la collection de l'écran et les insère dans une table interne « `ws_zamtcoll_new` » du format de la table des collections.
Dans la table « `ws_zamtcoll_old` » qui est de même format, sont stockées les données de collection avant qu'elles soient modifiées (cette table est vide en création).

J'utilise une BAPI déjà créée qui enregistre dans la BDD dans laquelle je place ces tables en paramètre d'entrée.
Lors de la création ou duplication d'une collection il y a aussi la table « `ws_zamtcollt` » à remplir, on y place le code collection et son libellé. On place aussi cette table en paramètre de la BAPI.

L'enregistrement des sous-collections.
Dans cette partie encore il m'a fallu gérer différents cas de figure. Le premier cas est celui où aucune sous-collection n'est renseignée. Dans cette situation je récupère le code de la collection, son libellé et les insère dans les champs respectifs de la table interne « `z_zpssscol` » dans laquelle je stock les informations de la sous-collection.
Si une ou des sous-collections sont renseignées, je les insère dans la table « `z_zpssscol` ».

Cette table est ensuite passée en paramètre de la BAPI servant pour l'enregistrement.

Le log de sauvegarde.

A la suite d'une sauvegarde le client désire que l'utilisateur soit prévenu de son bon fonctionnement. J'ai créé un pop-up récapitulant la collection et les données de la pyramide pour résumer une création ou une duplication.
Voici l'exemple d'un pop-up après la duplication d'une collection :

Conclusion

Pour conclure, ces 6 mois passés au sein d'Augusta Reeves ont été très enrichissants. Ils m'ont permis de mettre en application les connaissances acquises durant mon parcours scolaire.

Sur le plan personnel, le fait d'avoir travaillé et beaucoup échangé autour du projet m'a permis de développer mes capacités à travailler en équipe. L'ambiance de travail était agréable, il y avait un bon esprit d'équipe et des pratiques d'entraide qui ont favorisé les échanges. Ce stage m'a apporté une expérience professionnelle supplémentaire et m'ouvre les portes du monde du travail puisqu'il débouche sur un CDI.

Sur le plan professionnel ce stage a été aussi très bénéfique puisque je suis entré dans l'univers SAP, logiciel très utilisé de nos jours par tous types d'entreprises. J'ai pu apprendre de nouvelles technologies à travers l'utilisation du langage ABAP. Bien que la prise en main des logiciels nécessaires à mon projet ait été longue et parfois difficile, j'ai désormais de solides bases sur SAP.
Sur ce logiciel on sent qu'on progresse jour après jours en l'utilisant mais l'on sait qu'il nous reste toujours des choses à apprendre tellement il est complet, c'est pour moi un des principaux avantage de travailler avec SAP et une des raisons qui m'a donné envie d'en faire mon métier.

Outre le fait que mon stage s'est très bien déroulé personnellement et professionnellement c'est une des raisons pour lesquelles j'ai donné une suite favorable au contrat qui m'a été proposé.

Pour l'entreprise, mon travail a été intéressant car j'ai pu réaliser les tâches qui m'ont été confiées. Mon travail a contribué à l'avancée du projet de la mise en place de SAP à la SODIS mais également d'autres projets sur lesquels j'ai pu être amené à travailler en moindre mesure.

De plus le fait qu'un contrat me soit proposé prouve que mon travail a été apprécié ce qui représente une grande satisfaction personnelle.

Bibliographie

1. **Ferraud-Ciandet, Maurin.** Comment mettre en place et maintenir des développements spécifiques sur les solutions de SAP en toute pérennité ? 2013.

2. **Sanchez-Vaqué, Raphael.** Création d'une maquette d'application pour tablette Samsung. 2013.

3. http://fr.wikipedia.org/wiki/ABAP. *wikipedia.* [En ligne]