# Migration de la chaîne décisionnelle du calcul des taux d'usure et proposition d'une méthodologie de migration du logiciel SAS vers R

Ludovic Merville

le cnam

**CONSERVATOIRE NATIONAL DES ARTS ET METIERS PARIS**

---

# MEMOIRE

**présenté en vue d'obtenir**

**le DIPLOME D'INGENIEUR CNAM**

**SPECIALITE : Informatique**

**OPTION : Systèmes d'information**

**par**

**M. Ludovic MERVILLE**

---

Migration de la chaîne décisionnelle du calcul des taux d'usure et proposition d'une méthodologie de migration du logiciel SAS vers R

**Soutenu le 26 mars 2018**

---

**JURY**

**PRESIDENT :** - M. Eric Gressier-Soudan, Professeur au CNAM

**TUTEUR :** - Mme. Elisabeth Métais, Professeur au CNAM

**MEMBRES :** - Mme. Faten Atigui, Maître de Conférences au CNAM
- M. Laurent Landrea, Chef du Service des Projets et Maintenance Statistiques (Banque de France)
- M. Olivier Desrumaux, Chef adjoint du Service des Projets et Maintenance Statistiques (Banque de France)

# REMERCIEMENTS

De nombreuses personnes ont contribué à ce mémoire et je tiens à les remercier.

Pour commencer, merci au Professeur Elisabeth METAIS pour sa patience, ses relectures dans la bonne humeur et ses bons conseils. Le mémoire en a été grandement amélioré.

Je remercie M. Olivier DESRUMAUX, chef adjoint du Service et Projet de Maintenance Statistiques de m'avoir permis et soutenu à réaliser ce mémoire.

Je remercie également mes collaborateurs Mme Sabrina SOUCHETTE, Mme Emanuelle SOURMEY, M. Pascal GORIOT, M. Guillaume MARCHAND, M. Mehdi HAZAZETA, M. Jérémy ROUSSET et M. Yann AMAROUCHE pour leur contribution quotidienne et le plaisir de travailler ensemble.

Je remercie plus largement le CNAM pour sa qualité d'enseignement. Ces cinq années m'ont, entre autre, fait évoluer personnellement, m'ont permis d'acquérir de nombreuses connaissances et de rencontrer des personnes intéressantes de divers horizons. Cela a contribué et continue à rendre mon travail meilleur, m'a conféré une assurance dans mes connaissances, une forte capacité de recherche et d'obtenir une confiance et des responsabilités de plus en plus grandes de ma hiérarchie.

Je remercie ma conjointe Mme Hélène BRIAND pour son soutien de tous les jours et ses remarques pertinentes.

# LISTE DES ABREVIATIONS

TEG : Taux Effectif Global
TAEG : Taux Annuel Effectif Global
TESE : Taux effectif au sens étroit
EAI (ou en français IAE) : Enterprise Application Integration (ou en français Intégration d'Applications d'Entreprise)
OMS : Objets de Métier Spécifiques
ETL : Extract, Transform and Load
SAS : Statistical Analysis System
MOA : Maîtrise d'ouvrage
MOE : Maîtrise d'œuvre
MOM : Message-Oriented Middleware
XML : Extensible Markup Language
XSD : XML Schema Definition
DSS : Decision Support Systems (Système d'aide à la décision)
SGBD : Système de Gestion de Base de Données
OLAP : Online Analytical Processing
ODS : Operational Data Store
VABF : Vérification d'Aptitude Au Bon Fonctionnement
DGS : Direction Générale des Statistiques
DIMOS : Direction de l'Ingénierie et de la Maîtrise d'Ouvrage Statistiques
SPMS : Service des Projets et Maintenances Statistiques
OI : Organisation de l'Information
ACPR : Autorité de Contrôle Prudentiel et de Résolution
BCE : Banque Centrale Européenne
BCN : Banque Centrale Nationale
SEBC : Système Européen de Banques Centrales
UE : Union Européenne
SEC : Système Européen de Comptes
ONEGATE : Organisation Nouvelle des Échanges via un Guichet d'Alimentation et de Transfert vers l'Extérieur
ROSTAM : Réforme Obligatoire des STAtistiques Monétaires
SISMF : Système d'Information pour les Statistiques Monétaires et Financières
CIB : Corporate & Investment Bank

# GLOSSAIRE

Le **taux effectif global**, ou **taux annuel effectif global** est le taux d'intérêt fixé par la banque ou l'établissement de crédit. C'est la somme d'argent que vous devez payer en plus de la somme que vous empruntez effectivement (le capital).

Le **taux d'usure** correspond au taux (TAEG) maximum que tous les prêteurs sont autorisés à pratiquer lorsqu'ils accordent un crédit. L'instauration d'un tel seuil vise à protéger l'emprunteur d'éventuels abus.

Un **établissement de crédit** est un organisme financier qui pratique des opérations de banque à titre de profession habituelle.

Un **middleware** ou intergiciel est un logiciel tiers qui crée un réseau d'échange d'informations entre différentes applications informatiques.

En programmation informatique, le **test unitaire** (ou « T.U. », ou « U.T. » en anglais) ou test de composants est une procédure permettant de vérifier le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme (appelée « unité » ou « module »).

Le **XML** (Extensible Markup Language) est un langage de balisage extensible. Il permet donc de créer à l'infini de nouvelles balises pour identifier tout type de donnée.

# TABLE DES MATIERES

# TABLE DES FIGURES

# TABLE DES TABLEAUX

# 1 INTRODUCTION

L'évolution est partie intégrante de la vie et les projets informatiques ne font pas exception. La question de l'adaptation d'un système d'information, d'une application à une nouvelle découverte, à des nouvelles données, à de nouveaux outils, à de nouveaux besoins… est régulière. Les entreprises évoluent également par effet de contingence interne ou externe, volonté d'innovation, fusions ou acquisitions, introduction de nouveaux savoirs… Ces évolutions peuvent engendrer de complexes modifications du système d'information de l'entreprise. Une migration informatique est alors un événement courant.

Je travaille depuis mai 2015 à la Banque de France dans la Direction Générale des Statistiques (DGS) en tant que chef de projet SAS - R. Mon équipe est constituée de trois internes spécialisés en SAS (dont moi-même), quatre consultants externes spécialisés dans le décisionnel et un alternant d'école d'ingénieur informatique (Polytech Lille) pour une période de trois ans. Nous avons comme objectif d'aider à la réalisation de projets statistiques sous SAS et R et d'accompagner la DGS à une plus large utilisation du logiciel R. C'est dans ce contexte professionnel que j'ai réalisé ce projet.

Pour la Banque de France, la dernière réforme quinquennale imposée par la Banque Centrale Européenne a engendré une migration technique et une mutualisation d'une partie du système d'information de la DGS. Une partie de la chaîne décisionnelle du calcul des taux d'usure a été modifiée. Seule la collecte des données via le portail OneGate de la Banque de France n'est pas modifiée. Le stockage de ces données est migré dans un nouveau système d'information et la chaîne SAS de calcul des taux d'usure en SAS doit donc s'y adapter.

La problématique est alors de réaliser une migration partielle de la chaîne décisionnelle de l'usure. Etant le chef de projet pour la partie SAS de cette migration, j'ai alors réalisé les tâches suivantes :

- L'analyse des impacts du nouveau système d'information sur la chaîne SAS ;
- La rédaction du cahier des charges ;
- Le chiffrage des évolutions SAS ;
- Les spécifications techniques de l'adaptation de la chaîne de traitement ;
- L'implémentation en SAS ;
- La recette et superviser celle du métier pour la validation des résultats ;
- Le script de livraison de la partie SAS ;

- Les comptes rendus d'avancement de la partie SAS au comité de Pilotage ;
- Le choix du nouveau modèle de données du système d'information en collaboration avec la Direction de l'Organisation de l'Information (OI).

A la DGS, environ deux cent cinquante personnes (économistes, chargés d'études…) travaillent quotidiennement avec le logiciel SAS. Au total, environ six cent cinquante personnes qui utilisent SAS de façon régulière à la Banque de France.

L'arrivée à maturité et un usage croissant de logiciels open-source beaucoup moins coûteux que SAS conduit la Banque de France à privilégier cette classe de solutions. L'intégration de ces outils dans des environnements mutualisés renforcera la palette d'outils à disposition des analystes et développeurs pour la réalisation ou le prototypage d'outils analytiques. En pratique, ces environnements seront également une des solutions à disposition des analystes souhaitant migrer des productions (chaines statistiques, analyses, …) réalisées jusqu'à maintenant dans un environnement SAS. A la DGS, le choix a été fait de mettre en avant la solution R.

R est conçu pour les statistiques. Il offre des fonctions de visualisation des données et d'analyse avancées. Il bénéficie de mises à jour et d'enrichissements permanents des librairies de programmes (par opposition aux releases espacées de SAS) ainsi que d'une base de connaissances très riche maintenue par une communauté mondiale d'utilisateurs. Tout comme SAS, R peut lire des formats variés de données : fichiers Excel, bases des données externes (SQL serveur, Oracle…), fichiers de données SAS…

L'option retenue consiste à privilégier les outils développés avec le langage de programmation R, complétée le cas échéant par d'autres solutions comme Python. En contrepartie l'activité SAS est appelée à décroître graduellement. Ce qui signifie la reprise en R des programmes SAS déjà développés par la DGS et jugés essentiels pour l'activité de production ou d'étude.

La problématique est alors d'estimer le coût de migration d'un traitement et l'accompagnement utilisateur nécessaire. Dans cette optique, en tant que chef de projet, je suis chargé de :

- Proposer et réaliser une méthodologie d'estimation du coût de migration des programmes SAS vers le logiciel R ;

- Accompagner les utilisateurs de la DGS à la prise en main du logiciel R Studio (via un navigateur web) sur un serveur informatique LINUX partagé.

Après avoir exposé mon cadre de travail au sein de la Banque de France, je détaillerai ma problématique puis, j'expliquerai le fonctionnement de la chaine des taux d'usure. Je réaliserai ensuite une partie théorique afin de pouvoir discuter des décisions prises lors ce projet. Je terminerai par mes réponses apportées à la problématique.

# 2 PRÉSENTATION DE L'ENTREPRISE

Je travaille depuis mai 2015 à la Banque de France dans la Direction Générale des Statistiques (DGS) en tant que chef de projet SAS - R. Mon équipe est constituée de trois internes, quatre consultants externes spécialisés dans le décisionnel et un alternant d'école d'ingénieur informatique. Nous avons comme objectif d'aider la réalisation de projets statistiques sous SAS et R et d'accompagner la DGS à une plus large utilisation du logiciel R. C'est dans ce contexte professionnel que j'ai réalisé ce projet.

## 2.1 La Banque de France

Membre de l'Eurosystème, qui regroupe la Banque centrale européenne et les banques centrales nationales des pays ayant adopté l'euro pour monnaie, la Banque de France est une personne morale publique « sui generis », régie par les dispositions du code monétaire et financier. Les conditions dans lesquelles elle exerce ses missions sur le territoire national sont définies par le Contrat de service public.

Au sein de l'Eurosystème, les décisions sont centralisées au niveau du Conseil des gouverneurs (qui regroupe les membres du Directoire de la BCE (Banque Centrale Européenne) et les gouverneurs des BCN (Banque Centrale Nationale) de la zone euro. La mise en œuvre des décisions est décentralisée au niveau des BCN.

Par ailleurs, afin de tenir compte du fait que plusieurs pays de l'Union européenne (UE) n'ont pas encore adopté l'euro, le Système Européen de Banques Centrales (SEBC) rassemble la BCE et toutes les banques centrales de l'UE au sein du Conseil général.

Les trois grandes missions de la Banque de France sont la stratégie monétaire, la stabilité financière et le service économique à la collectivité.

La définition de la stratégie monétaire, extraite du site https://www.banque-france.fr/politique-monetaire/presentation-de-la-politique-monetaire/la-strategie, est la suivante :

> « L'Eurosystème a annoncé, le 13 octobre 1998, une stratégie de politique monétaire, c'est-à-dire, une définition quantifiée de l'objectif de stabilité de prix et une description du cadre d'analyse permettant au Conseil des Gouverneurs de prendre ses décisions de politique monétaire qui est la suivante : « La stabilité des prix est

définie comme une progression sur un an de l'indice des prix à la consommation harmonisé (IPCH) inférieure à 2% dans la zone euro. ». »

Pour la stratégie monétaire, la Banque de France prépare et met en œuvre les décisions prises par le Conseil des gouverneurs. Elle effectue donc des études, des analyses, des prévisions, des enquêtes statistiques…pour obtenir un diagnostic le plus fiable possible au niveau macroéconomique et financier.

La BCE définit la stabilité financière de la façon suivante : (https://www.ecb.europa.eu/ecb/tasks/stability/html/index.fr.html)

> « La stabilité financière est l'expression d'une situation empêchant l'émergence de risques systémiques. Le risque systémique se définit comme le risque que la fourniture de produits et services financiers par le système financier soit entravée à un point tel que la croissance économique et le bien-être pourraient s'en trouver considérablement affectés. »

En matière de stabilité financière, la Banque de France a une double responsabilité de protection et de surveillance :

- Elle est en charge du renforcement de la réglementation et de la prévention des risques ainsi que de la sécurité des dépôts des épargnants ;
- Elle assure avec l'ACPR (Autorité de Contrôle Prudentiel et de Résolution) la supervision des entreprises du secteur financier (établissements bancaires, entreprises d'assurance et mutuelles), veille au bon fonctionnement des systèmes de paiement et des infrastructures de marché et procède régulièrement à l'évaluation des risques et vulnérabilités du système financier.

En terme de services, la Banque de France s'occupe entre autre de :

- La tenue du compte du Trésor public ;
- Des dossiers de surendettement ;
- La cotation des entreprises pour les PME ;
- L'imprimerie des billets en euro ;
- La surveillance de la qualité de la monnaie fiduciaire et procède au retrait des coupures en mauvais état.

La Banque de France publie un rapport économique, commentant les évolutions économiques, monétaires et financières et les politiques engagées en France, en Europe et dans le monde, et un rapport d'activité présentant les actions de la Banque de France dédiées à ses missions et à sa responsabilité sociétale, ainsi que sa gestion financière et les comptes de l'exercice.

Pour gérer ses fonctions, la Banque de France est constituée d'un ensemble de directions (cf. Figure 1) toutes rattachées au Conseil Général. La Banque de France est présidée par son Gouverneur nommé par décret du président de la République. Depuis 1er novembre 2015, le Gouverneur est M. François VILLEROY DE GALHAU.

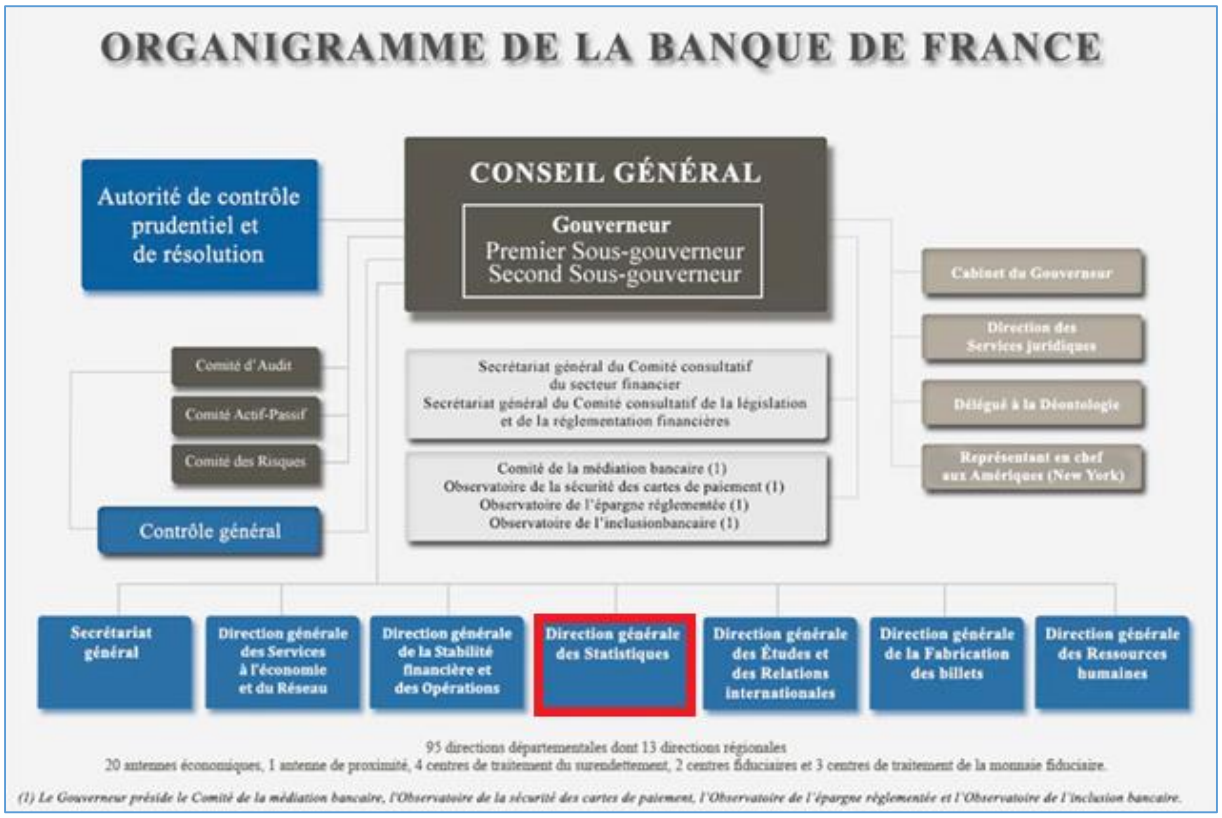


*Figure 1 : Organigramme de la Banque de France, source : intranet Banque de France*

L'intégralité du mémoire a été effectuée à la Direction Générale des Statistiques. Je détaille donc son organisation et ses principales missions.

## 2.2 LA DIRECTION GENERALE DES STATISTIQUES

La Direction Générale des Statistiques (DGS) est présidée par M. Jacques FOURNIER. En quelques chiffres c'est :

- Environ 300 employés ;
- Quatre directions (cf. Figure 2) ;
    - Direction de l'Ingénierie et de la Maîtrise d'Ouvrage Statistiques (DIMOS) ;
    - Direction des Statistiques Monétaires et Financières (DSMF) ;
    - Direction de la Balance des Paiements (DBDP) ;
    - Direction des Enquêtes et Statistiques Sectorielles (DESS).
- Quatorze services.

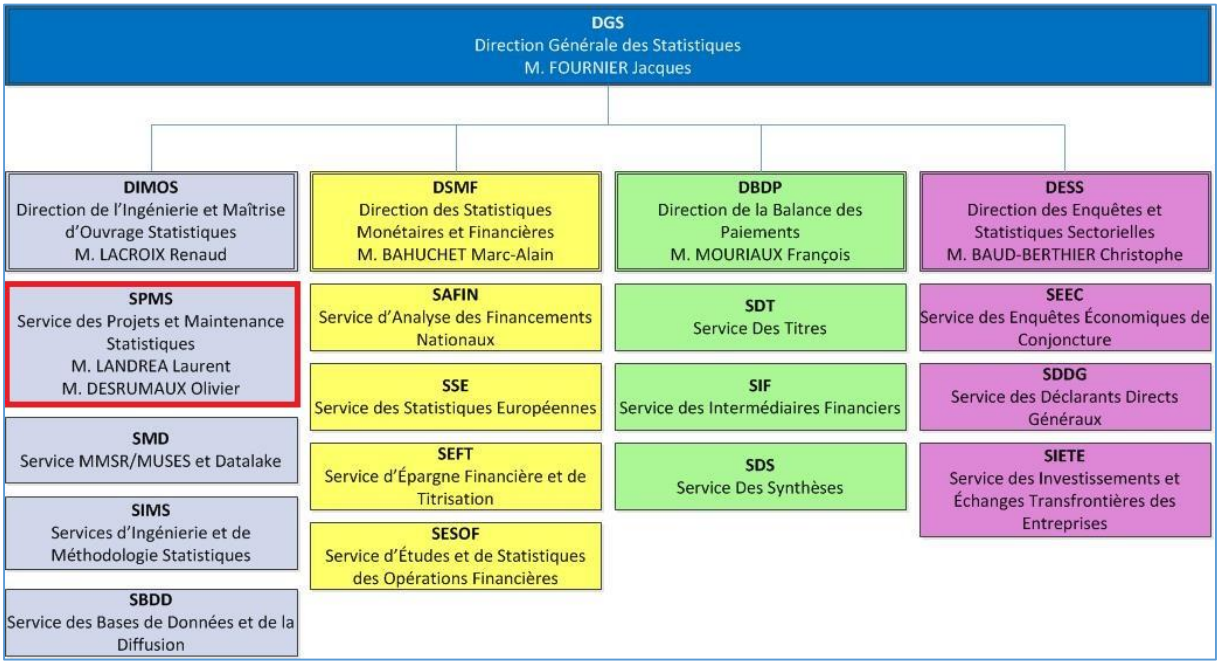


*Figure 2 : Organigramme de la DGS*

Elle collecte, exploite, analyse et diffuse des statistiques dans un grand nombre de domaines. Il s'agit en particulier :

- Des statistiques monétaires et financières : crédits, dépôts, agrégats monétaires, calcul des encours et des taux des crédits aux ménages et aux entreprises ;
- Des enquêtes de conjoncture : au niveau national, pour l'industrie et les services et par secteur (industrie, services marchands…) ;
- De la balance des paiements : transactions courantes, position extérieure de la France, émission et détention de titres de toute nature ;
- Du calcul des taux réglementés (usure, livret A…).

La DGS assure le secrétariat de plusieurs instances présidées par le Gouverneur comme l'Observatoire de l'épargne réglementée, l'Observatoire de l'inclusion bancaire, etc.

La DGS contribue ainsi à alimenter la prise de décision du gouvernement de la Banque de France, notamment en matière de politique monétaire et financière. Ces données sont à destination de la BRI, FMI, OCDE, BCE…

Le cadre du mémoire s'est déroulé à la DIMOS.

## 2.3 La Direction de l'Ingenierie et Maitrise d'Ouvrage Statistiques

La Direction de l'Ingénierie et Maîtrise d'Ouvrage Statistiques (DIMOS) est dirigée par M. Renaud LACROIX et a pour mission de :

- Rationaliser et renforcer les fonctions « support » de l'activité statistique ;
- Piloter la maîtrise d'ouvrage des projets et l'exploitation métier des applications ;
- Développer l'expertise méthodologique ;
- Diffuser et recevoir l'information sous diverses formes.

La DIMOS met à disposition du grand public un ensemble de séries statistiques (taux, crédit, comptes nationaux financiers…) sur le portail statistique WebStat (http://webstat.banque-france.fr/fr/).

En complément de ce portail, la DIMOS gère pour les chercheurs un accès gratuit aux données individuelles. Cela est fait par une « Open Data Room » (https://www.banque-france.fr/statistiques/acces-aux-donnees-granulaires/open-data-room).

Dans cette direction, je travaille pour le SPMS (Service des Projets et Maintenances Statistiques). Je détailler donc ses principales missions.

## 2.4 Le Service des Projets et Maintenances Statistiques

Au sein de la DIMOS, le SPMS comprend 35 personnes. Le chef de service est M. Laurent LANDREA et l'adjoint du chef de service est M. Olivier DESRUMAUX. Le SPMS est responsable, en étroite coordination avec les services utilisateurs et les services informatiques, de la maîtrise d'ouvrage de nombreux projets et maintenances de la DGS. Il assure le pilotage de la plateforme de collecte OneGate, supervise les référentiels de données de la DGS et réalise l'intégration progressive des applicatifs de la DGS (production

des statistiques monétaires, de la balance des paiements, des enquêtes de conjoncture…). Le SPMS fournit une assistance et contribue aux développements statistiques SAS et R.

## 2.5 ENVIRONNEMENT INFORMATIQUE DE TRAVAIL

La DGS utilise majoritairement le logiciel SAS pour le traitement et l'analyse des données. Une plus grande utilisation de R fait partie des objectifs de la DGS. La DGS accède à de multiples formats de bases de données comme ORACLE, SQL SERVER et DB2.

### 2.5.1 Le logiciel SAS

SAS (Statistical Analysis System) est un logiciel statistique et de traitement de données avec un langage de programmation propriétaire dont le début a été dans les années 1960 à l'Université de North Carolina aux Etats-Unis. La société SAS a été créé en 1976 et ce logiciel permet de :

- Consulter, traiter, exploiter des tables SAS, des bases de données et des fichiers (texte, Excel, XML…) ;
- Effectuer des calculs de statistiques, de l'analyse de données, du Data Mining ;
- Représenter des données sous forme de rapports et de graphiques ;
- Développer des applications Web ;
- …

On peut classer SAS dans la catégorie des ETL (Extract-Transform-Load).

La base du langage SAS est :

- Les étapes dites « data » servent à la manipulation des tables de données. On peut y créer des variables, réaliser des calculs, etc. Le résultat est stocké dans une table physique qui facilite la validation du traitement.
- Les « procédures ou proc » sont des traitements programmés par SAS. Cela peut être le calcul de moyenne (proc means), de fréquence (proc freq), de tests statistiques (khi-deux, loi normale…), des coefficients de corrélation (proc corr) … Il existe la « PROC SQL » qui permet de réaliser du langage SQL.

SAS permet de gérer les droits d'accès à l'aide de son serveur de métadonnées. Chacune des directions de la DGS a son propre serveur de métadonnées pour administrer ses propres droits. Lors d'une connexion à SAS, le serveur de métadonnées vérifiera le bon « login / mot de passe » et les différents accès autorisés. En effet, SAS permet la création de « bibliothèque » qui permet d'accéder directement à un répertoire ou une base de données externe. Lors de la création d'une bibliothèque il est possible d'ajouter dans les métadonnées des droits d'accès pour les utilisateurs. Ainsi, uniquement certains utilisateurs auront accès à la bibliothèque soit en écriture soit en lecture. Ces droits sont uniquement des droits dédiés à SAS.

Dans le cadre du mémoire, c'est le logiciel SAS Entreprise Guide V6.1 qui a été utilisé.

### 2.5.2 Le langage R

D'après le site https://www.r-project.org :

> « R est une langue et un environnement Open Source pour l'informatique statistique et les graphiques. C'est un projet GNU (GNU's Not UNIX) qui est similaire à la langue et l'environnement S qui a été développé chez Bell Laboratories (anciennement AT & T, maintenant Lucent Technologies) par John Chambers et ses collègues.
>
> R fournit une grande variété de statistiques (modélisation linéaire et non linéaire, tests statistiques classiques, analyse des séries chronologiques, classification, regroupement, etc.), des techniques graphiques et est hautement extensible. »

R permet d'installer des bibliothèques (ou package) développées et mises à disposition par la communauté R pour intégrer de nouvelles fonctions statistiques, graphiques…

R est un logiciel Open-Source et la Banque de France le propose en téléchargement via son site d'application. Mais, afin de permettre aux personnes de travailler avec plus de puissance, dans un espace partagé et d'accéder aux différentes bases de données externes, la DGS a initié la mise en place d'une plateforme R STUDIO sur un serveur LINUX.

Le cadre professionnel du projet étant désormais exposé, j'en explique la problématique.

# 3 PROBLEMATIQUE

La problématique est de mettre en œuvre deux aspects de « migration » informatique :

- Une migration partielle de la chaîne décisionnelle du calcul des taux d'usure ;
- Une proposition de méthodologie de migration des traitements SAS vers des traitements R.

## 3.1 MIGRATION DE LA CHAINE DECISIONNELLE DU CALCUL DES TAUX D'USURE

La migration partielle de la chaîne décisionnelle du calcul des taux d'usure fait partie d'un projet interne à la DGS du nom de ROSTAM (Réforme Obligatoire des STAtistiques Monétaires). Elle est motivée par l'obligation pour la Direction des Statistiques Monétaires et Financières de se conformer à deux évolutions réglementaires européennes :

- La réforme quinquennale des statistiques monétaires et financières édictée par la BCE ;
- L'entrée en vigueur d'un nouveau Système Européen de Comptes (SEC 2010) défini dans le règlement du Parlement européen et du Conseil de l'Union Européenne n° 549/2013 du 21 mai 2013, qui implique de procéder à un changement de base de comptabilité nationale.

La problématique est alors de répondre à cette migration partielle de la chaîne décisionnelle de l'usure et donc plus précisément à trois parties distinctes :

- L'envoi des données de collecte dans le nouveau système d'information (ROSTAM) ;
- La migration du système d'information de stockage de ces données et proposition d'un nouveau modèle de données ;
- L'évolution des programmes SAS des calculs des taux d'usure suite aux changements ci-dessus.

Il y a donc plusieurs équipes dont chacune est responsable des évolutions de sa partie. Je suis le chef de projet en charge de la problématique des évolutions SAS des calculs des taux d'usure. Je dois proposer une solution délivrant les résultats de l'usure conformes ainsi qu'une nouvelle interface de lancement. Étant un utilisateur avancé de la chaîne de l'usure, je participe également à l'analyse de la solution du nouveau modèle de données.

En parallèle, je dois répondre à une problématique de méthodologie de migration de SAS vers R.

## 3.2 PROPOSITION DE METHODOLOGIE DE MIGRATION DE SAS VERS R

A la DGS, environ deux cent cinquante personnes (économistes, chargés d'études…) travaillent quotidiennement avec le logiciel SAS. Au total, à la Banque de France, environ six cent cinquante personnes utilisent SAS de façon régulière. Il y a donc un fort historique de programme SAS de plus d'une quinzaine d'années.

La Banque de France s'oriente vers une plus large utilisation du logiciel R et une diminution graduelle de celle du logiciel SAS. Les nouveaux projets de traitement de données seront donc plus fréquemment réalisés en R et les programmes SAS existants seront migrés au fur et à mesure en R. Dans une migration de logiciel, il y a le coût d'installation/licence des nouveaux outils, de formation et d'accompagnement du personnel, de montée en compétence, de migration des programmes existants dans la nouvelle technologie, etc. C'est sur ce dernier point que mon expertise a été requise.

Ma direction m'a posé la problématique de trouver une méthode estimant le coût de migration des programmes SAS vers R. Cette connaissance permettra d'avoir un ordre de grandeur et de savoir s'il faut faire appel à des ressources externes ou si les équipes en place suffisent pour migrer un projet dans la contrainte de temps souhaitée.

Donc, à la différence de la problématique de la migration de la chaîne décisionnelle de l'usure qui est un projet collaboratif avec plusieurs équipes, la problématique de méthodologie est un projet individuel.

La problématique étant désormais exposée je vais expliquer le fonctionnement du calcul du taux d'usure.

# 4 PRESENTATION DU TAUX D'USURE

## 4.1 DEFINITION

Le taux d'usure fixe le taux de crédit maximum aux particuliers selon différentes catégories (cf. Tableau 7 en annexe). Il est calculé et publié trimestriellement par la Banque de France et plus précisément par la DGS. La définition issue du site de la Banque de France (https://particuliers.banque-france.fr/votre-banque-et-vous/credits-aux-particuliers/le-taux-dusure) est la suivante :

> « Le taux de l'usure est défini par la loi (articles L. 313-3 à 6 du code de la consommation et article L. 313-5 du code monétaire et financier). Le taux de l'usure est le taux d'intérêt maximum qu'un prêteur a le droit d'appliquer à un prêt aux particuliers, au moment où le prêt est consenti. Un prêt est considéré comme usuraire lorsqu'il est consenti à un TEG qui excède du tiers le taux effectif moyen pratiqué au cours du trimestre précédent. »

La collecte de l'usure recense, de manière exhaustive, les nouveaux contrats de crédit libellés en euros, conclus avec les particuliers, les sociétés non financières, les entrepreneurs individuels, les institutions sans but lucratif au service des ménages et les administrations publiques locales, résidant en France ou non-résidents EMUM (États Membres de l'Union Monétaire).

Il est utilisé par la Banque de France afin de concourir à l'élaboration des statistiques de taux d'intérêt sur les contrats nouveaux requises par le règlement BCE/2009/7 du 31 mars 2009 de la Banque centrale européenne.

En outre, ce tableau de données est utilisé pour collecter les données permettant de calculer le taux de l'usure conformément au décret n° 90/506 du 25 juin 1990.

Sa description est disponible sur le site internet de la Banque de France à l'adresse suivante : http://www.banque-france.fr/fr/statistiques/declarants/modalites-techniquesetablissements-credits.htm

## 4.2 LA COLLECTE DES DONNEES

Les données relatives à l'usure sont collectées trimestriellement via le tableau appelé M_CONTRAN (Contrat Nouveau). Un échantillon d'établissements de crédit ou de guichets

d'établissements de crédit est soumis à cette collecte. L'échantillon est remis à jour annuellement mais peut être modifié trimestriellement en fonction des événements qui affectent les guichets de l'échantillon. La liste des établissements de crédit assujettis à la remise de ce tableau à compter de l'échéance de février 2010 est disponible sur le site internet de la Banque de France à l'adresse suivante :

http://www.banque-france.fr/fr/statistiques/declarants/modalites-techniques-etablissements-credits.htm

Les différentes banques de crédit communiquent ces données via le portail OneGate de la Banque de France.

### 4.2.1 Le portail OneGate

ONEGATE (Organisation Nouvelle des Échanges via un Guichet d'Alimentation et de Transferts vers l'Extérieur) est un portail mutualisé de collecte par internet d'informations statistiques, financières, administratives et prudentielles. Il permet l'utilisation de formats et de modalités de collectes adaptés aux profils des déclarants et à la nature des informations demandées (cf. Figure 3).

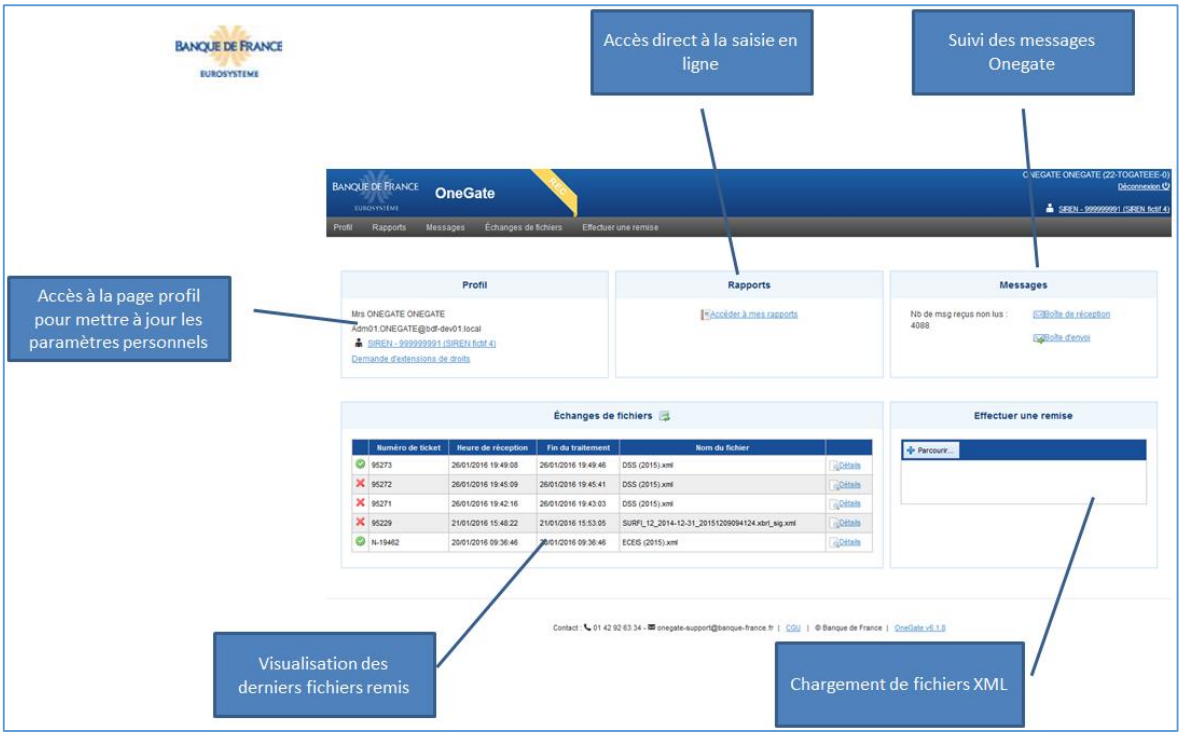


*Figure 3 : Fenêtre d'accueil du portail OneGate*

Le portail permet également de :

- Suivre la production ;
- Suivre les remises des remettants ;
- Consulter les données saisies par les remettants.

Le remettant peut, soit saisir manuellement, soit déposer un fichier XML (respectant la norme imposée par OneGate via un XSD). OneGate réalise plusieurs contrôles (conformité, de structure et de référentiel). Ne travaillant pas sur la partie de remise et contrôle, je ne la détaille pas. Je vais décrire le fichier XML car son contenu est utilisé pour le calcul des taux d'usure.

### 4.2.2 Le fichier de collecte M_CONTRAN

Chaque fichier de collecte ayant passé les contrôles avec succès, se compose de deux parties :

- Une première partie administrative ;
- Une seconde partie spécifique aux données collectées.

#### *4.2.2.1 La partie administrative*

La partie administrative (cf. Figure 4) contient des informations relatives aux données échangées (institution, domaine et identification du déclarant) et sa structure est la suivante :

```
<Administration creationTime="2010-03-26T09:29:25.154+01:00">
	<From declarerType="CIB">12345</From>
	<To>BDF</To>
	<Domain>MCO</Domain>
	<Response>
		<Email>mail@mailpro.com</Email>
		<Language>FR</Language>
	</Response>
</Administration>
```

*Figure 4 : Exemple de fichier XML M_CONTRAN - partie administrative*

La balise <Administration creationTime> correspond à la date de réception de la remise dans OneGate.

La balise <From declarerType> correspond à l'identification du déclarant. C'est un format caractère de longueur 5.

La balise <To> correspond au destinataire de la collecte et a pour valeur "BDF".

La balise <Domain> correspond à l'identifiant du domaine de la collecte et a pour valeur « MCO ».

La balise <Response> contient l'adresse email (<Email>) de l'émetteur et le langage (<Language>) de l'avis de dépôt qui pour M_CONTRAN vaut toujours « FR ».

Toutes les balises de la partie administration sont obligatoires.

#### *4.2.2.2 La partie spécifique aux données collectées*

Cette partie contient les données par ligne de crédit. Elles sont identifiées selon cinq cas (cinq formulaires) :

- MCO1 : opérations avec les particuliers ;
- MCO2 : opérations avec les sociétés non financières ;
- MCO3 : opérations avec les entrepreneurs individuels ;
- MCO4 : opérations avec les institutions sans but lucratif au service des ménages ;
- MCO5 : opérations avec les administrations publiques locales.

Pour chacun de ces formulaires, les données collectées diffèrent sur quelques balises XML mais la grande majorité des informations restent communes.

Ci-dessous (cf. Figure 5), un extrait de données collectées sur deux formulaires.

```
<Report date="2016-12" code="MCO">
	<Data form="MCO2">
		<Item>
			<Dim prop="SCT">MCO2</Dim>
			<Dim prop="ID_GUI">00001</Dim>
			<Dim prop="RFLICR">12345678901234</Dim>
			<Dim prop="INS_FI">200</Dim>
			<Dim prop="MT_CRDT">120000</Dim>
			…
			<Dim prop="SIREN">123456789</Dim>
		</Item>
	</Data>
	<Data form="MCO3">
		 <Item>
			<Dim prop="SCT">MCO3</Dim>
			<Dim prop="ID_GUI">00001</Dim>
			<Dim prop="RFLICR">95684532154568</Dim>
			<Dim prop="INS_FI">420</Dim>
			<Dim prop="MT_CRDT">245000</Dim>
			…
			<Dim prop="SIREN">987654321</Dim>
		</Item>
	</Data>
</Report>
```

*Figure 5 : Exemple de fichier XML M_CONTRAN - partie spécifique*

La balise <Report date= "AAAA-MM" code=""> comprend :

- La période de remise au format AAAA-MM, par exemple :
    - Échéance de janvier 2017 : 2017-01 ;
    - Échéance de juillet 2017 : 2017-07.
- Le code correspondant à l'identifiant du rapport, ayant pour valeur "MCO".

La balise <Data form> correspond à l'identifiant du formulaire et a pour valeur "MCO1", "MCO2", "MCO3", "MCO4" ou "MCO5".

La balise <Item> correspond à la déclaration d'une ligne de crédit et sa description est fonction du formulaire.
La description du fichier XML pour le formulaire MCO1 est en annexe (cf. Tableau 8). Les autres formulaires ne sont pas présentés car leurs structures sont très similaires.

Le contenu de ce fichier XML est stocké dans la base de données SISMF (Système d'Information pour les Statistiques Monétaires et Financières) et au terme de ce projet, dans ROSTAM. La chaîne SAS de calcul du taux d'usure requête donc ce système d'information pour sa production trimestrielle.

## 4.3 ORGANISATION DE LA CHAINE SAS DU CALCUL DU TAUX D'USURE

La chaîne du calcul du taux d'usure consiste aux fonctionnalités suivantes :

- Extraction des lignes de crédit collectées dans le tableau M_CONTRAN ;
- Contrôle des informations et relance des déclarants sur les anomalies avérées ;
- Écrêtage suivant des distributions par critère ;
- Calcul des statistiques de l'usure.

Elle est réalisée avec le logiciel SAS. Elle est découpée en quatre étapes (cf. Figure 6).

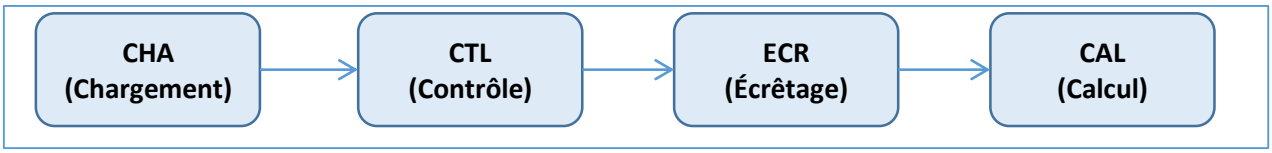


*Figure 6 : Organisation fonctionnelle de la chaîne SAS du calcul du taux d'usure*

# 5 CONNAISSANCES THEORIQUES

Ce chapitre est consacré à l'explication de la théorie des différents aspects techniques abordés dans ce rapport. Ces éléments permettront de discuter les choix effectués.

## 5.1 EAI

Pour l'échange des données entre le serveur SAS et le serveur ROSTAM, l'OI (Organisation de l'Information) a proposé d'utiliser EAI. Ne connaissant pas ce principe, j'ai réalisé une recherche à ce sujet.

### 5.1.1 Objectifs

La définition de EAI extraite de Wikipédia est la suivante :

> « L'intégration d'applications d'entreprise ou IAE (en anglais enterprise application integration, EAI) est une architecture intergicielle permettant à des applications hétérogènes de gérer leurs échanges. On la place dans la catégorie des technologies informatiques d'intégration métier (business integration) et d'urbanisation. Sa particularité est d'échanger les données en pseudo temps réel. »

EAI est un des moyens de répondre au besoin de communication entre différentes applications hétérogènes du Système d'Information qui a plusieurs objectifs [1]:

- Présenter à l'utilisateur final une vision unifiée de l'information gérée par les différentes applications ;
- Masquer la complexité du système d'information et donc faciliter la prise de décision ;
- Résoudre le problème de la cohérence entre des systèmes qui s'entrecroisent.

Lorsqu'il y a peu d'applications à intégrer, une approche simple à mettre en place est celle de la connexion point à point. Pour un nombre n d'applications à connecter entre elles, il est nécessaire de mettre en place n (n-1)/2 tuyaux de communication [2].

Mais dès qu'il y a un plus grand nombre d'applications, alors ce mode devient complexe. Pour 5 applications, il faut alors 20 tuyaux de communication. La comparaison avec le « syndrome du plat de spaghettis » évoquée par John Rusby est alors appropriée :

« A badly structured program is likened to a plateful of spaghetti : if one strand is pulled, then the ramifications can be seen at the other side of the plate where there is a mysterious turbulence and upheaval ».

Si une application évolue alors un grand nombre d'applications sont concernées. L'illustration ci-dessous (cf. Figure 7, extrait de [3]) montre la différence de complexité de lecture et de compréhension entre une communication entre chaque application et une communication centralisée avec EAI.

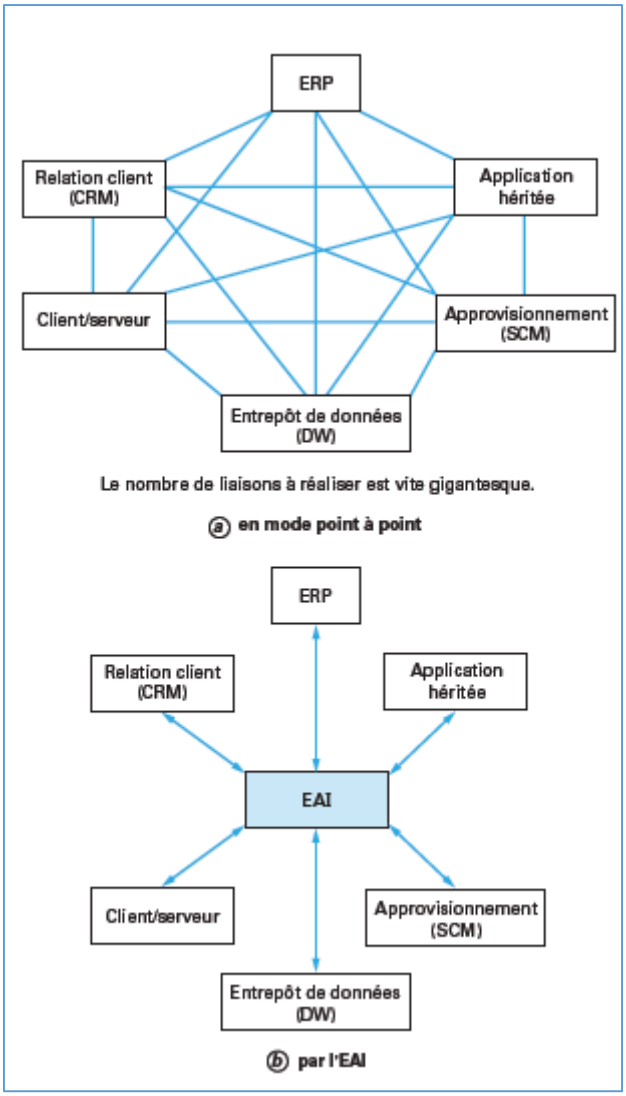


*Figure 7 : Communication entre applications, source Techniques de l'ingénieur, EAI H2915*

EAI fonctionne principalement avec trois types de fonctions [4] :

- Transport ;
- Connecteurs ;
- Le Moteur d'intégration avec la transformation et le routage.

### 5.1.2 Le transport

Les services de transport assurent la livraison des données aux applications via le moteur d'intégration. Plus généralement, une solution d'EAI doit savoir se connecter à tout middleware existant et doit proposer des passerelles entre les middlewares hétérogènes présents dans l'entreprise. L'intégration s'opère ainsi à tous les niveaux : applications, données et middlewares.

Le service de transport réalise la communication entre les applications. Il repose sur un middleware de communication (propriétaire ou éditeur partenaire) qui peut être asynchrone ou synchrone [5].

Dans un système asynchrone, une application A émettant un message vers une application B peut continuer à travailler sans avoir reçu le message de retour de B. Il y a alors une gestion de file d'attente de message. L'application B stockera le message jusqu'à ce qu'il soit traité. Cette solution utilise les « Message Oriented Middleware » (MOM) comme IBM MQ ou MSMQ. Plusieurs caractéristiques peuvent être définies sur le message :

- La priorité dans la file d'attente des messages ;
- La garantie que le message sera traité de façon unique ;
- La suppression du message à partir d'une certaine date s'il n'est pas encore traité ;
- La confidentialité ou non du contenu du message ;
- …

Dans un système synchrone, une application A émettant un message vers une application B doit attendre le message de retour de B pour continuer à travailler. On doit alors utiliser :

- Les Object Request Broker (ORB) dont la norme proposée par l'OMG (Object Management Group) est CORBA (Common Object Request Broker Architecture, http://www.omg.org/spec/CORBA/);
- Le protocole HTTP (HyperText Transfer Protocol).

Un outil EAI, est également en charge de réaliser les transformations sur les informations portées par les messages afin d'adapter les données de l'émetteur aux formats gérés par le récepteur.

### 5.1.3 Les connecteurs applicatifs

Les connecteurs applicatifs (ou adaptateurs applicatifs) permettent la communication entre la plateforme EAI et les applications du système d'information. Les connecteurs permettent de se connecter aux applications et gèrent l'authentification, les transactions, et les droits d'accès. Ils extraient les données de l'application émettrice et communiquent le résultat à l'application réceptrice. Ils doivent être non intrusifs, c'est-à-dire qu'il n'y a pas de programmation ajoutée dans les applications émettrices et réceptrices. Il existe différents type de connecteurs [6] :

- SGBD (Oracle, SQL Server, DB2…) ;
- ORB ;
- ERP (SAP, Oracle Applications…) ;
- …

Ces connecteurs sont, soit achetés chez l'éditeur de l'EAI, soit développés en spécifique avec un kit de développement (SDK, Software Development Kit) fourni par l'éditeur.

Il existe deux types de connecteurs, les statiques et les dynamiques.

Les connecteurs statiques peuvent être :

- Des connecteurs légers pour lesquels un développement sera nécessaire pour les faire fonctionner ;
- Des connecteurs riches pour lesquels une interface graphique pourra remplacer le développement réalisé.

Dans les deux cas, si un changement s'opère (par exemple sur le SGBD) il faudra adapter manuellement les connecteurs.

Les connecteurs dynamiques ou intelligents sont notamment capables de s'adapter automatiquement à un changement sur le SGBD (ou autre).

Lorsqu'un connecteur transmet des données, elles sont nommées Objets de Métier Spécifiques (OMS). Les transformations des messages entre applications sont réalisées dans le moteur d'intégration.

### 5.1.4 Le moteur d'intégration

Le moteur d'intégration désigne le message broker qui transforme et transmet les messages entre les applications. Les données en sortie d'une application et en entrée d'une autre peuvent avoir des formats différents. Par exemple la date peut être de multiples formats. Elle peut s'écrire de façon « mm/yyyy » dans une application et « dd/mm/yyyy » dans une autre. Le message broker gère ces transformations. L'étape de « mapping des données » permet alors d'établir ces règles. Ce mapping reçoit donc un OMS et le transforme en OM (Objets de métier).

Un des intérêts techniques à réaliser les transformations dans le message broker est de les centraliser dans un outil dédié. Il est possible de réaliser des enrichissements d'informations comme des agrégations, ajouts de champs, de l'ordonnancement de tâches exécutées sur des applications avec des plateformes différentes, de la gestion de processus de rythme différent (exemple : mettre une base de données à jour uniquement lorsqu'un ou plusieurs processus seront terminés) …

Le routage peut être :

- Statique, soit défini par le type du flux ;
- Dynamique, soit défini par le contenu du flux.

Dans les deux cas il peut être un :

- One-to-one, soit une communication d'une application A vers une application B ;
- Many-to-many, soit une communication d'une ou plusieurs applications (fusion d'informations, calcul, agrégation…) vers une ou plusieurs autres applications.

Les règles de transformation et de routage peuvent être contenues dans un référentiel de métadonnées qui dictera les règles à appliquer lors de l'arrivée d'un message.

La gestion du routage peut également être réalisée en fonction des processus métier. On parle alors de BPM (Business Process Management).

### 5.1.5 Les différentes architectures

#### *5.1.5.1 Architecture Hub and Spoke*

L'architecture Hub and Spoke (cf. Figure 8, extrait de [3]) centralise l'ensemble des communications avec un hub unique. Le référentiel des règles de routage et de transformation est donc également centralisé. L'administration de la plateforme en est alors simplifiée. Un des inconvénients majeurs est que si, le hub n'est plus disponible, alors aucune intégration ne sera possible. De plus la gestion de charge est complexe à gérer au vu de la centralisation.

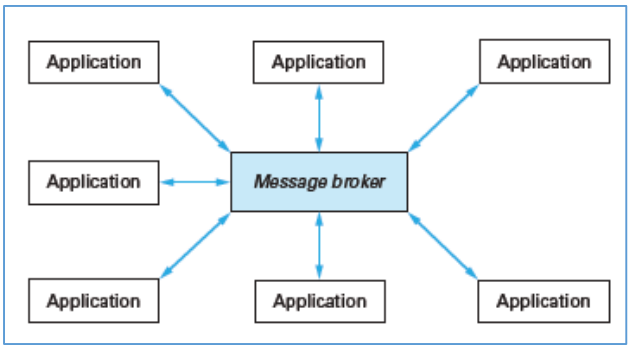


*Figure 8 : Architecture Hub and Spoke, source : Techniques de l'ingénieur, Enterprise Application Integration : EAI, réf H2915*

#### *5.1.5.2 Architecture Multi-Hub*

L'architecture Multi-hub (cf. Figure 9, extrait de [3]) est un dérivé de « Hub and Spoke ». Elle peut être mise en place lors d'un plus grand nombre d'utilisation d'applications. Il faudra alors dupliquer les référentiels de transformation et de routage sur chacun des hubs. C'est d'ailleurs un de ses inconvénients de maintenance.

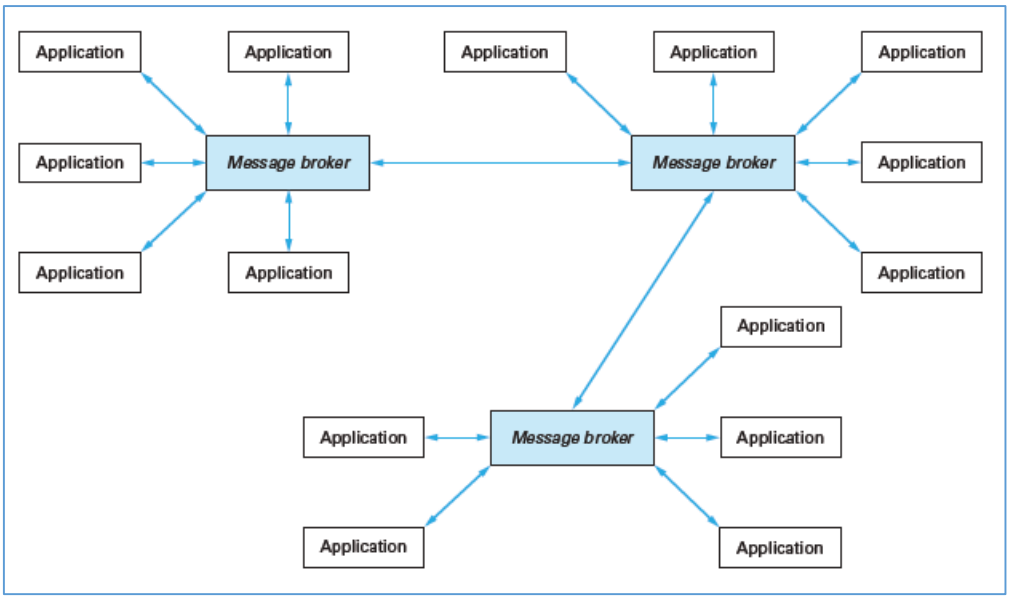


*Figure 9 : Architecture Multi-Hub, source : Techniques de l'ingénieur, Enterprise Application Integration : EAI, réf H2915*

#### *5.1.5.3 Architecture Network Centric*

L'architecture Network Centric (ou bus applicative) (cf. Figure 10, extrait de [3]) distribue les services sur plusieurs serveurs. C'est donc une version décentralisée. Les référentiels de règles de transformation et de routage sont implémentés sur l'ensemble des nœuds (point de connexion à une application). L'émission d'un message d'une application est alors traitée avec son propre nœud. Un des avantages est que la charge de traitement est répartie sur l'ensemble des nœuds et non sur un seul (solution Hub and Spoke). La complexité de mise en œuvre est un des inconvénients par rapport aux solutions « Hub ».

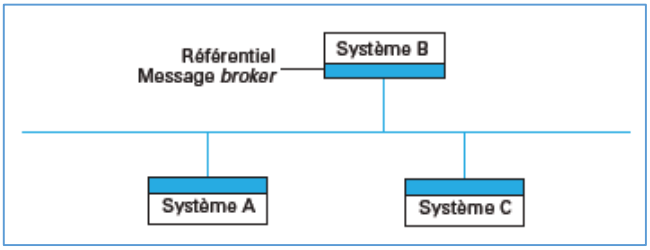


*Figure 10 : Architecture Network Centric, source : Techniques de l'ingénieur, Enterprise Application Integration : EAI, réf H2915*

Pour mettre en place une plateforme EAI, il existe donc plusieurs architectures possibles. Il existe également différents types d'intégration des données.

### 5.1.6 Les différentes intégrations

Il existe trois types d'intégration de données (cf. Figure 11) dont la complexité de mise en œuvre varie [7].

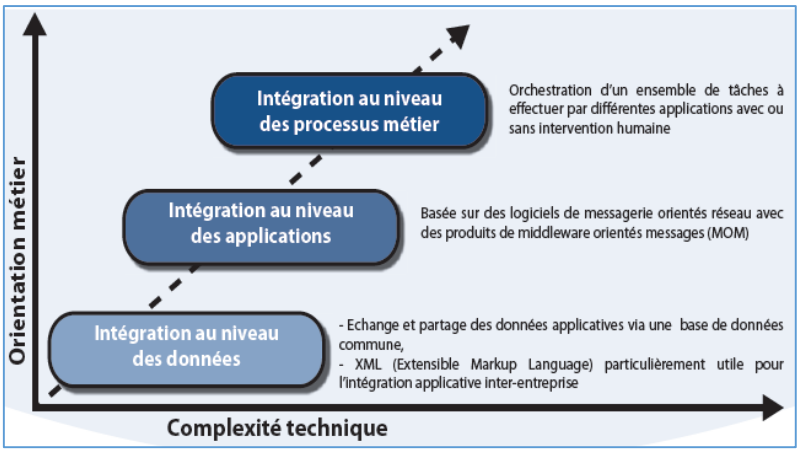


*Figure 11 : Les différents types d'intégration EAI, source cours « Stratégies de développement des Systèmes d'Information Opérationnels de l'entreprise », Bernard Espinasse, supinfo, http://slideplayer.fr/slide/498193/*

### *5.1.6.1 Intégration au niveau des données*

L'intégration au niveau des données permet d'obtenir des bases de données plus cohérentes. Lors d'une modification/création/suppression de données, les bases de données impactées sont mises à jour (cf. Figure 12, extrait de [6]). Au besoin, les règles métiers de vérification et de transformation peuvent être appliquées.

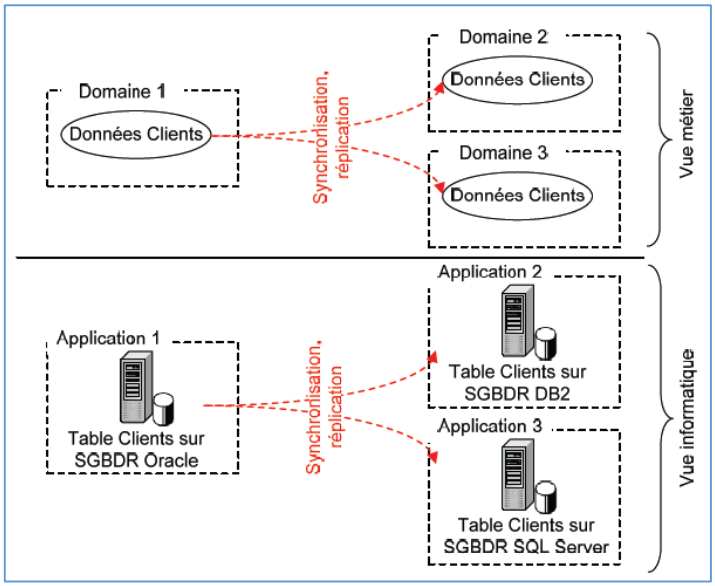


*Figure 12 : EAI : Intégration au niveau des données, source CNRS "Panorama d'une infrastructure EAI"*

Sur la figure ci-dessus (cf. Figure 12, extrait de [6]), l'EAI permet, lors d'une création d'un client (ou autre), de mettre à jour sur les autres bases de données « Clients » du système d'information.

### *5.1.6.2 Intégration au niveau des applications*

L'intégration au niveau des applications permet de récupérer les règles métiers des applications en passant par des API métiers. L'EAI a donc le rôle de communication avec les API (cf. Figure 13, extrait de [6]).

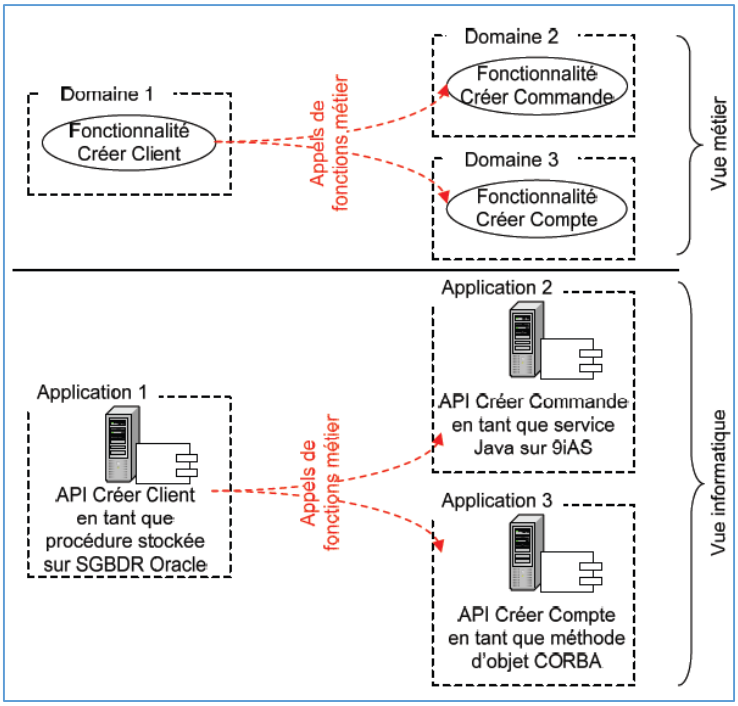


*Figure 13 : EAI : Intégration au niveau des applications, source CNRS "Panorama d'une infrastructure EAI"*

Sur la figure ci-dessus (cf. Figure 13, extrait de [6]), l'EAI communique avec les API métier afin de mettre lors d'une création d'un client (ou autre), et de créer également une commande et un compte.

### *5.1.6.3 Intégration au niveau des processus métier*

À la différence des deux précédentes intégrations qui sont « techniques », cette dernière est fonctionnelle. L'intégration des processus métier est dirigée par ces derniers. On l'appelle aussi BPI (Business Process Integration). L'EAI aura pour rôle d'orchestrer l'intégration des données et de suivre l'intégration (bonne création du client, délai de validation…) (cf. Figure 14, extrait de [6]).

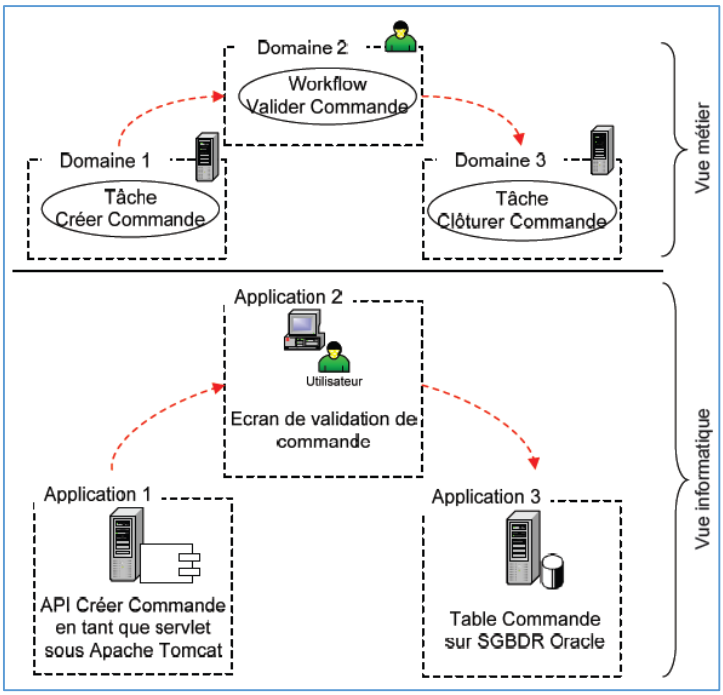


*Figure 14 : EAI : Intégration au niveau processus métier, source CNRS "Panorama d'une infrastructure EAI"*

Sur la figure ci-dessus (cf. Figure 14, extrait de [6]), l'EAI orchestre de façon temporelle les actions depuis la création d'une commande jusqu'à sa clôture.

Nous allons désormais étudier la théorie sur la chaîne décisionnelle.

## 5.2 LA CHAINE DECISIONNELLE

Une décision ne peut être prise sans information, sans donnée, sauf à le faire au hasard. Tout au long de sa vie le cerveau humain enregistre des informations que nous utilisons chaque jour pour prendre des décisions : « Il fait 30°, est-ce que je prends mon manteau ? ». Pour prendre cette décision banale nous sélectionnons uniquement certaines informations face à la multitude dont nous disposons. Paul Valéry a écrit : "Que de choses il faut ignorer pour agir". Je prendrai cette citation dans le sens « décisionnel ». En effet, pour prendre une décision, il faut ignorer toutes les informations inutiles. C'est le cas des entreprises qui doivent, elles aussi, avoir des informations pertinentes sur lesquelles se baser afin d'être compétitives, augmenter le profit, trouver un cœur de cible, analyser les changements climatiques…Le nombre de données augmentent sans cesse (internet, réseaux sociaux, internet des objets…) et l'informatique décisionnelle a pour objectif de les utiliser pour en extraire de l'information utile.

En 1994 William Inmon a défini le concept d'entrepôt de données ou en anglais Data Warehouse :

> « Un entrepôt de données est une collection de données thématiques, intégrées, non volatiles et historisées, organisées pour le support à la prise de décision. »

L'entrepôt de données est alors le pilier central de l'informatique décisionnelle.

### 5.2.1 Le schéma décisionnel

La figure ci-dessous (cf. Figure 15) explique le schéma décisionnel depuis la collecte d'information jusqu'à sa restitution à l'utilisateur final. Il est avec un ODS (Operational Data Store) qui peut ou pas faire partie de l'architecture. Nous examinerons plusieurs possibilités dans le chapitre « 5.2.8 Diverses architectures possibles ».

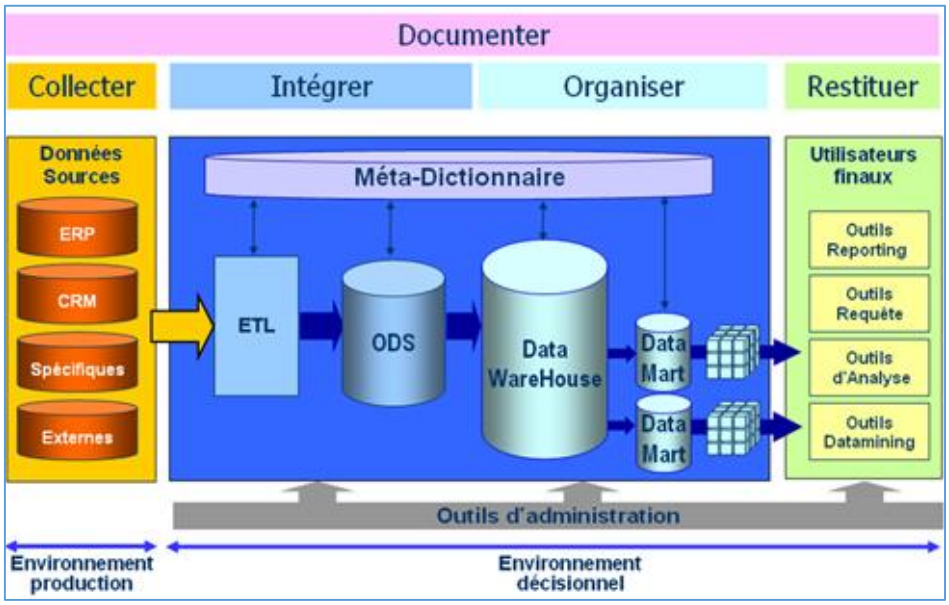


*Figure 15 : L'entrepôt de données dans le schéma décisionnel, source http://perso.univ-lyon1.fr/haytham.elghazel/BI/presentation.html*

Dans le schéma décisionnel on distingue différentes étapes :

- Les sources de données ;
- L'extraction (ETL) ;
- L'intégration (ODS).

### 5.2.2 Les données sources

Les données sources (ou les données de production) sont extraites afin que le « travail décisionnel » ne se réalise pas directement dessus. En effet, les traitements d'analyses peuvent être lourds et donc ralentirent ceux de production et peut être les mettre en péril.

Les données utiles à l'entreprise pour fonctionner, également appelées données transactionnelles, servent dans les systèmes opérationnels ou OLTP (On-Line Transaction Processing). Ces données sont dédiées aux métiers opérationnels de l'entreprise. Par exemple, les données des commandes reçues le jour-même, et dirigées vers la personne vérifiant leur contenu. Elles n'ont pas vocation à être historisées, mais bien uniquement à faire fonctionner l'entreprise quotidiennement. C'est par ailleurs l'une des raisons d'extraire et de stocker ces informations dans un autre système plus décisionnel.

Ces systèmes décisionnels ou OLAP (On-Line Analytical Processing) sont orientés pour les décideurs de l'entreprise, pour les études, etc. En reprenant l'exemple de la commande du jour, les données seraient l'ensemble des commandes reçues et non uniquement celle du jour.

De par leur nature de fonctionnement et d'objectif, il existe plusieurs différences entre ces deux systèmes qui sont récapitulées dans le tableau ci-dessous (cf. Tableau 1, extrait de [8]).

| | **OLTP (bases transactionnelles de production)** | **OLAP (cubes analytiques)** |
|---|---|---|
| **Utilisateur** | Collaborateur, cadre opérationnel | Cadre fonctionnel, décideur |
| **Fonction** | Saisie journalière | Aide à la décision |
| **Base de données** | Orientée application (ERP) | Orientée métier |
| **Données** | Dynamique | Historique |
| **Usage** | Répété | À la demande (*ad hoc*) |
| **Accès** | Lecture/écriture | Lecture seule (écriture de simulation possible) |
| **Unité de travail** | Transaction (insertion/suppression, mise à jour). Langage SQL | Requête complexe hiérarchique. Langage MDX |
| **Nb enregistrements utilisés** | Quelques enregistrements | Millions d'enregistrements |
| **Nb utilisateurs** | Centaines | Dizaines |
| **Volume de la Base** | GB | TB |

*Tableau 1 : Comparaison OLTP versus OLAP, source Bertrand Burquier : BUSINESS INTELLIGENCE AVEC SQL SERVER 2008 : Mise en œuvre d'un projet décisionnel*

### 5.2.3 Les métadonnées

Les métadonnées, soit les données sur les données, permettent de mieux comprendre les données. Une métadonnée peut être, par exemple, une coordonnée GPS sur une photo permettant de savoir exactement où elle a été prise ou une étiquette sur une boîte de conserve pour connaître son contenu, etc. Les métadonnées données sont dites l'ADN de l'entrepôt de données qui définissent tous les éléments et comment ils fonctionnent ensemble [9].

Dans le décisionnel, les métadonnées doivent pouvoir décrire le processus d'alimentation et le contenu du Data Warehouse. Dans [10], Bill Inmon les décrit comme « un index du contenu de l'entrepôt de données ». Elles permettent, en les interrogeant, de retrouver les données plus rapidement, de savoir lesquelles seront utiles, de connaître les transformations opérées…

La figure ci-dessous (cf. Figure 16, extrait de [10]) montre l'importance des métadonnées pour le mapping entre les données du système opérationnel et celles de l'entrepôt de données. En effet, une donnée peut changer de nom, subir des transformations, etc. Sans cette « description », il est alors plus difficile de retrouver l'origine d'une donnée et donc de bien la comprendre.

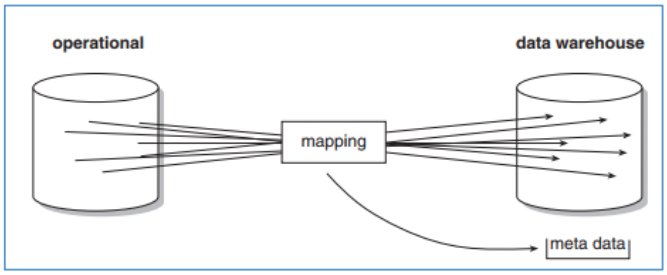


*Figure 16 : Mapping entre données opérationnelles et le Data Warehouse, extrait du livre Bill Inmon : Building the Data Warehouse*

Un entrepôt de données est évolutif. Les données sources peuvent évoluer mais surtout les besoins métiers peuvent changer. En effet, si un modèle donne de bons résultats, les utilisateurs désireront ajouter des informations afin de pouvoir améliorer leurs analyses. Ils émettront aussi des nouveaux besoins. Les métadonnées stockeront alors les différentes évolutions de l'entrepôt de données (cf. Figure 17, extrait de [10]).

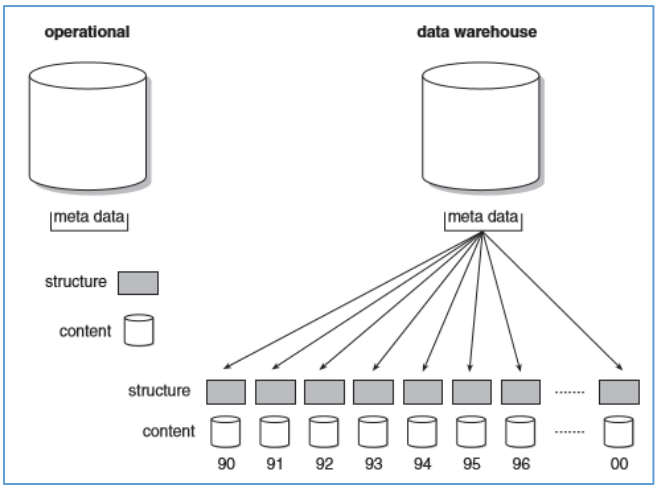


*Figure 17 : Une historisation de la structure du Data Warehouse à l'aide des métadonnées, extrait du livre Bill Inmon : Building the Data Warehouse*

Les métadonnées du schéma décisionnel peuvent être classées en trois catégories [11] (cf. Figure 18, extrait de [11]) :

- Le métier ou le business (utilisateurs finaux interrogeant les données) : elles décriront le contenu de l'entrepôt de données afin de le connaître le mieux possible ;
- Technique : elles décriront la structure des tables, les droits d'accès, les index, les partitions, les règles de transformation, etc ;
- De processus d'intégration : elles décriront afin de connaître l'ensemble des flux d'intégration, les règles, l'heure de début d'une tâche, les CPU utilisés, etc. Elles sont destinées aux responsables du processus d'intégration de l'entrepôt de données. Elles seront indispensables pour améliorer le processus d'intégration.

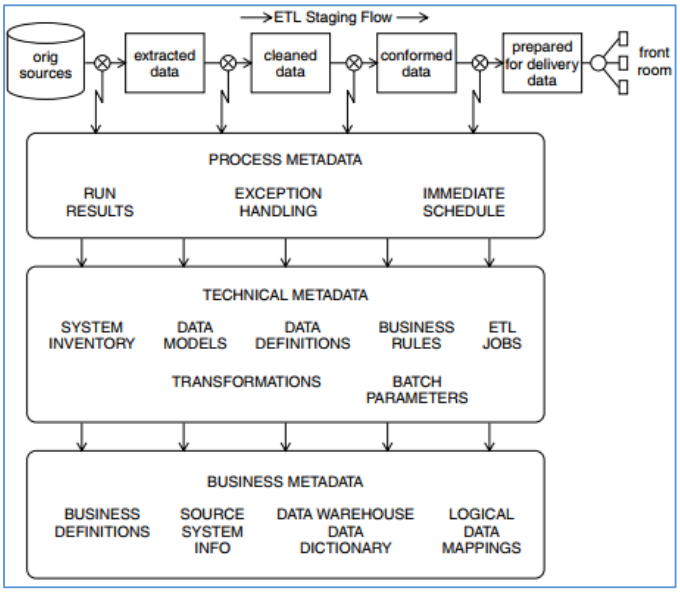


*Figure 18 : Les métadonnées du schéma décisionnel, extrait du livre The Data Warehouse ETL Toolkit, Ralph Kimball et Joe Caserta*

Les métadonnées sont donc utilisées sur l'ensemble du schéma décisionnel et potentiellement par de multiples outils différents. Par exemple, l'extraction de données peut être réalisée par deux ETL différents SAS et TALEND. Il faut donc que les métadonnées soient standardisées pour que chaque outil puisse les utiliser.

Une des principales spécifications de normalisation proposée par l'OMG (Object Management Group) est la CWM (Common Warehouse Metamodel). La spécification propose l'utilisation du XMI (XMI - XML Metadata Interchange) qui est fondée sur le XML. La spécification courante est disponible à l'adresse suivante : http://www.omg.org/spec/CWM/.

### 5.2.4 L'extraction

L'extraction est la première étape qui sélectionne les données nécessaires à l'entrepôt de données et donc celles nécessaires aux besoins métiers actuels ou futurs. L'extraction peut être réalisée avec un ETL, ce qui est présenté sur le schéma (cf. Figure 15). Il est également possible d'extraire avec d'autres méthodes, comme avec des interfaces ou des outils de réplications du SGBD.

Selon les besoins de l'entreprise, l'extraction et donc le rafraîchissement des données de l'entrepôt de données et des magasins de données, peuvent avoir différentes périodicités. Une mise à jour, le soir, afin d'éviter de consommer de la ressource en journée est une option souvent retenue.

Une fois les données extraites, l'intégration des données peut être réalisée.

### 5.2.5 L'intégration

Dans ce document, l'étape d'intégration inclut le nettoyage et la transformation de données.

Il y a deux points importants dans les données sources :

- Elles peuvent être hétérogènes : outre le format de stockage (fichier, web, base de données…) une représentation d'un même objet peut être différente selon la source :

    - Le sexe d'un individu peut être stocké sous forme textuel « Homme – femme » ou « H – F » ou « M – F » … ou sous forme numérique « 0 – 1 » ou « 1 – 2 » … (cf. Figure 19, extrait de [10]) ;
    - Le prix d'un produit peut être en TTC (toute taxe comprise) ou HT (hors taxe) ;
    - Les valeurs des ventes peuvent, par exemple, être en euro ou dollar en millier ou en million ;
    - Etc.
- Elles peuvent contenir des erreurs :
    - Saisir un zéro supplémenataire dans un nombre ;
    - Saisir un nombre en millier alors qu'il devrait être en million ;
    - Mal orthographier une ville, un pays ;
    - Une ville avec un code postal erroné ;
    - Etc.

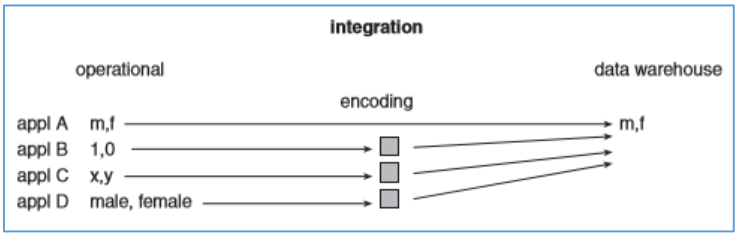


*Figure 19 : Choix de transformation de données entre les sources et l'entrepôt, extrait du livre Bill Inmon : Building the Data Warehouse*

L'étape de nettoyage et de transformation est donc essentielle pour la qualité des données et donc pour celui de l'entrepôt.

Pour uniformiser les informations, il est possible de créer des tables de paramétrages, de réaliser du code, des fonctions de conversion…Cela peut paraître, basique mais il ne faut pas oublier une transformation ou la faire dans le mauvais sens.

Le plus difficile est la correction des erreurs. Il faut d'abord les identifier. Exemple : une table de référence stockant l'ensemble des valeurs possibles pour un champ texte. Pour une ville, cela serait l'ensemble des noms des villes. Si une ville n'est pas dans cette table, alors elle sera donc en erreur. Par la suite, comment traiter l'erreur détectée ? Il y a plusieurs options :

- Manuelle : Une personne peut remplacer le nom erroné de la ville grâce aux autres informations. La méthode est sûre mais couteuse. ;
- Ontologique :t

- Dans [12], Thomas Gruber, défini l'ontologie de la façon suivante « An ontology is an explicit specification of a conceptualization ». Une ontologie définit une spécification formelle d'une conceptualisation partagée.
- La figure ci-dessous (cf. Figure 20, extrait de [13]), définit une ontologie sur les personnes d'un domaine scolaire. Dans cet exemple, si on reçoit d'une source la valeur « Professeurs » et d'une autre la valeur « Responsable de cours », alors on sait que tous les deux sont « Personnel ».

- Prioriser les sources d'information : Si un conflit apparaît entre plusieurs sources, on en privilégie une ;
- Etc.

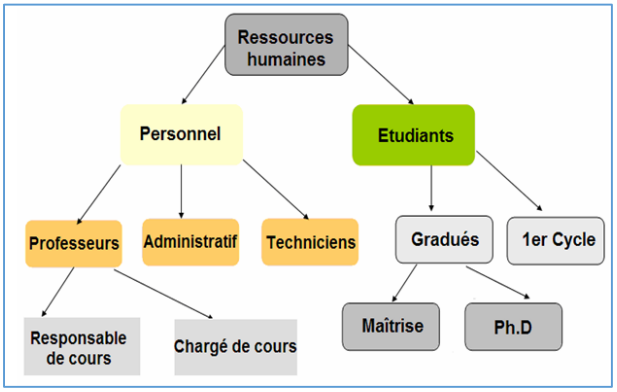


*Figure 20 : Exemple d'ontologie sur les personnes d'un domaine scolaire, extrait du mémoire « Vers une approche web sémantique dans les applications de gestion de conférences » de Mestiri Mohamed*

L'intégration peut également servir à réaliser des calculs, des agrégats afin d'optimiser les recherches ou tout simplement donner un résultat agrégé.

Les données étant désormais nettoyées et uniformisées elles peuvent être intégrées à l'entrepôt de données.

### 5.2.6 L'entrepôt de données : Data Warehouse

Bill Inmon définit l'entrepôt de données de la façon suivante :

> « Un entrepôt de données est une collection de données thématiques, intégrées, non volatiles et historisées, organisées pour le support à la prise de décision. »

L'entrepôt de données est thématique soit orienté vers les sujets majeurs de l'entreprise, à la différence des données opérationnelles qui sont, elles, autour des applications fonctionnelles de l'entreprise.

L'entrepôt de données est intégré. Comme vu dans le chapitre « 5.2.5 L'intégration » l'entrepôt a de multiples sources. Une même donnée peut alors avoir différentes notations, différents formats, etc. Des données peuvent être incohérentes, des conversions ou des agrégats sont sans doute également nécessaires. En ce sens, l'entrepôt doit être intégré.

L'entrepôt de données est non-volatile. Une donnée dans un entrepôt de données n'est jamais « mise à jour » directement. Dans ce cas on procède à un nouvel enregistrement afin de conserver la trace de sa valeur initiale. Ce qui est différent du système opérationnel, où une donnée peut être modifiée directement, voire même supprimée par exemple quand une commande est terminée (cf. Figure 21, extrait de [10]).

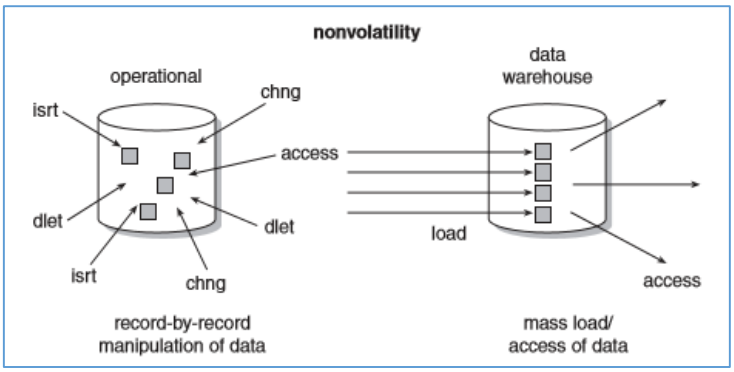


*Figure 21 : La non volatilité dans l'entrepôt de données, extrait du livre Bill Inmon : Building the Data Warehouse*

L'entrepôt de données est historisé. Les données sont donc liées à une variable temps. Il sera alors possible de connaître la chronologie des évènements. L'entrepôt de données stockera les données sur plusieurs années voire dizaines d'années. Pour certains domaines, comme le climat, le besoin d'historisation est quasi illimité.

L'entrepôt de données contient un grand volume d'informations dont le métier n'en sollicite qu'une partie. La mise en place d'un « magasin de données » (data mart) répond souvent à ce besoin de sélection.

### 5.2.7 Le magasin de données : Datamart

Un magasin de données est un sous-ensemble de l'entrepôt de données. Il répond à un besoin métier particulier. Il peut donc avoir un magasin de données pour l'équipe marketing et un autre pour l'équipe de vente.

Le magasin de données contient les données atomiques, c'est-à-dire les données de granularités les plus fines afin, au besoin, de pouvoir effectuer des recherches sur le cause

d'un évènement [11]. Il peut également contenir des données agrégées comme la somme des ventes par vendeur et par mois.

Dans le magasin de données, la volumétrie étant plus réduite que dans l'entrepôt, les temps d'accès sont normalement plus rapides (s'il est correctement construit).

### 5.2.8 Diverses architectures possibles

#### *5.2.8.1 Architecture à magasins matérialisés et entrepôt virtuel*

Précédemment, nous avons vu que le datamart se source depuis le data warehouse. Les approches de Ralph Kimball et Bill Inmon s'opposent sur ce sujet. Pour Ralph Kimball, le data warehouse est « logique », il n'existe que virtuellement. Il se compose de l'intégralité des datamarts qui eux sont « physiques ». Pour Bill Inmon le data warehouse est physique et les datamarts, physiques eux aussi, se sourcent de ce dernier. Il peut donc y avoir plusieurs conceptions sur le « data warehouse ».

#### *5.2.8.2 Architecture par médiateur*

L'architecture par médiateur est une intégration virtuelle des données [14]. L'entrepôt de données est virtuel. Il n'est pas stocké physiquement. Les données restent dans les sources et sont requêtées par l'intermédiaire de médiateurs et d'adaptateurs (ou wrappers). L'utilisateur requête le médiateur qui aura pour tâche de trouver l'information avec l'aide des adaptateurs (cf. Figure 22, extrait de [15]). Cette architecture permet d'interroger des données toujours à jour, même si les mises à jour sont très fréquentes.

Le médiateur définit ce que l'on appelle un schéma global. Il contiendra alors des vues abstraites sur les sources de données. Les utilisateurs interrogent ce schéma global. Les adaptateurs traduisent les requêtes dans le langage des sources afin de pouvoir extraire l'information de ce que l'on appelle schéma local.

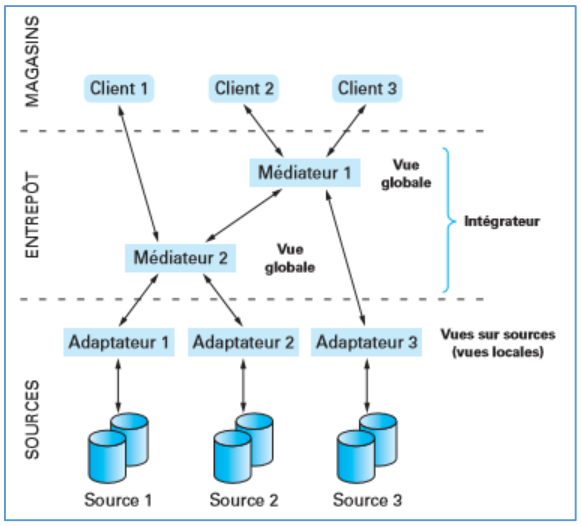


*Figure 22 : Architecture par médiateur, extrait de C. Chrisment, G. Pujolle, F. Ravat, O. Teste et G. Zurfluh, «Entrepôts de données,» Techniques de l'ingénieur - H3870, 2005*

Les requêtes écrites pour le médiateur sont traduites dans un langage défini pour l'appel de l'adaptateur. L'adaptateur interroge alors à son tour les données sources en écrivant des nouvelles requêtes adaptées à ces données. Un format de sortie est défini pour l'envoi des réponses de l'adaptateur vers le médiateur.

Pour ce type d'architecture, il existe plusieurs approches dont les deux principales sont le GAV (Global As View) et le LAV (Local As View).

L'approche GAV définit le schéma global comme une vue sur le schéma local [16]. L'exemple ci-dessous (cf. Figure 21, extrait de [17]) montre, avec cette approche, la création des vues abstraites du schéma global (Films et Articles) à partir des sources du schéma local (Source1.Film, Source2.Film, Source3.Critiques et Source3.Film).

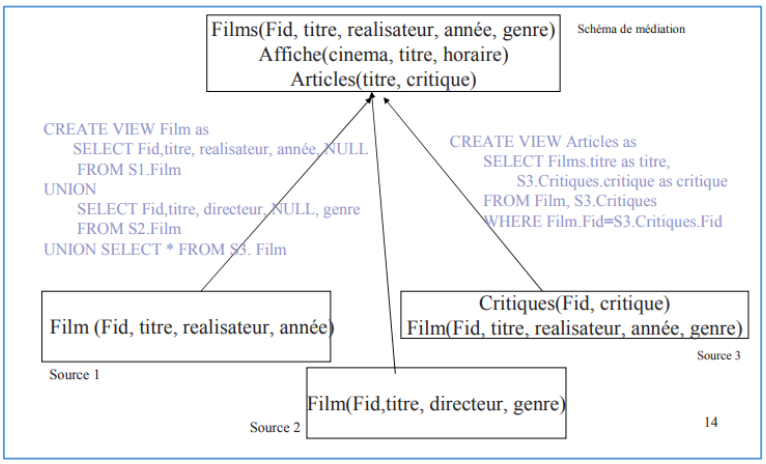


*Figure 23 : Architecture par médiation - exemple d'approche GAV, source Anne Doucet, Cours "Médiateurs", http://www-poleia.lip6.fr/~doucet/CoursBDIA/Cours5.pdf*

Un des principaux avantages de cette solution est la réécriture assez simple des requêtes depuis le médiateur vers le schéma local. En effet, les vues abstraites du schéma global sont réalisées à partir des vues du schéma local.

Un des inconvénients principaux est l'ajout de sources de données. Il faut alors réécrire le schéma global car il se base sur les sources.

L'approche LAV définit le schéma local comme une vue sur le schéma global [16]. C'est le raisonnement opposé à l'approche GAV. L'exemple ci-dessous (cf. Figure 21, extrait de [17]) montre, avec cette approche, la création des vues du schéma local (Source1.Film, Source2.Film et Source3.Critique) à partir du schéma global.

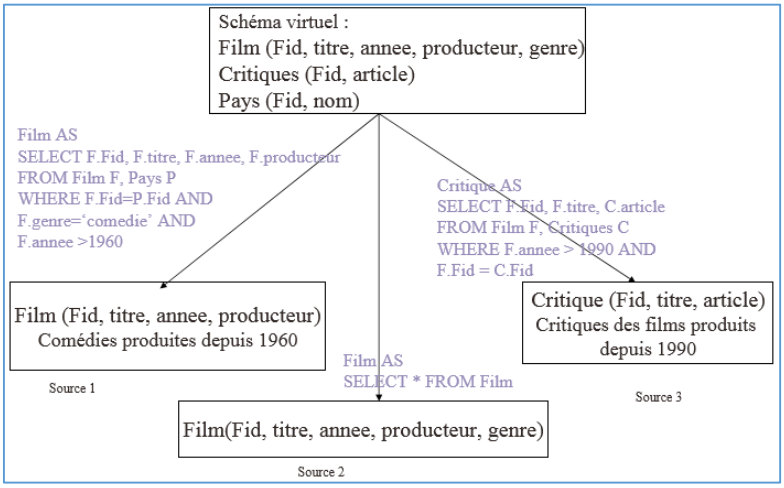


*Figure 24 : Architecture par médiation - exemple d'approche LAV, source Anne Doucet, Cours "Médiateurs", http://www-poleia.lip6.fr/~doucet/CoursBDIA/Cours5.pdf*

Un des avantages principaux de cette solution est l'ajout d'une source de données pour laquelle il suffit d'écrire une nouvelle requête. Ceci n'a alors pas d'impact sur le schéma global.

Un des inconvénients principaux est la complexité de la réécriture des requêtes entre le médiateur et le schéma local. En effet, les vues de ce dernier sont définies comme des vues du schéma global.

#### *5.2.8.3 Architecture avec ODS*

Il peut également avoir plusieurs choix sur le processus de nettoyage et d'intégration des données. Ce dernier pouvant être complexe, une zone de travail dédiée peut être nécessaire. Comme le montre la Figure 15, un ODS est un espace de stockage où les données

issues des sources sont nettoyées et intégrées avant d'être envoyées dans l'entrepôt. Les données n'y sont conservées que le temps de la résolution de tous les problèmes.

Nous avons donc vu le côté organisationnel de la chaîne décisionnelle. Nous allons désormais nous intéresser aux modèles de stockage de données.

### 5.2.9 Modèle OLAP

Le modèle OLAP (OnLine Analytical Processing) est un modèle multidimensionnel, défini par Edgar Frank Codd en 1993, dans un objectif d'analyse de données [18]. C'est est un cube ou hypercube qui permet l'analyse d'un évènement.

#### *5.2.9.1 Le cube OLAP*

Le cube OLAP (cf. Figure 25) stocke un évènement nommé « fait » qui peut être un achat de voiture, un appel, le téléchargement d'une application, etc. Je prendrai comme exemple le fait d'une vente d'un téléphone portable avec forfait pour expliquer la théorie du cube.

Le fait est expliqué par des axes d'analyse, soit « dimensions ». Dans notre exemple, il pourrait y avoir comme dimension :

- Client : Caractéristiques du client ;
- Point de vente : Caractéristiques du point de vente ;
- Téléphone : Caractéristiques du téléphone ;
- Abonnement : Caractéristiques de l'abonnement ;
- Temps : Le calendrier de l'année 2000 à 2099.

Les dimensions peuvent être hiérarchisées afin de réaliser des agrégations comme pour la dimension « Temps » qui serait « année – semestre – trimestre – mois – jour ». Une agrégation du nombre de vente de téléphone portable est alors disponible par an, semestre…

L'intersection des dimensions définit une cellule. Dans la figure ci-dessous (cf. Figure 25), l'intersection du client C2, du produit B et du mois d'août définit la cellule entourée en rouge.

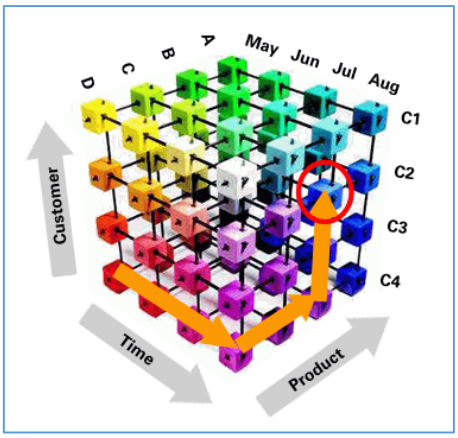


*Figure 25 : Exemple de cube OLAP à trois dimensions, source http://www.oracle.com/technetwork/articles/sql/11g-dw-olap-100058.html*

Une cellule contient les indicateurs ou mesures ou métriques ou variables selon les préférences de nomination. Il est possible, et c'est d'ailleurs le cas en général, de stocker plusieurs indicateurs en même temps. Dans notre exemple, les indicateurs sont :

- Le nombre de téléphones vendus ;
- Le nombre de mois d'engagement du forfait ;
- Le stock de téléphones.

Chaque mesure permet de calculer des agrégats dont la plus courante est la somme. Mais il est possible de calculer la moyenne, le minimum, le maximum, etc. Pour ce choix, il faut être vigilant au « type » de la mesure. Il existe trois catégories de mesure :

- **Mesure additive** : La mesure peut être additionnée sur toutes les dimensions. Dans notre exemple, la mesure « Nombre de téléphones vendus » est additive. Peu importe la dimension, sa somme a toujours du sens (cf. Figure 26).
- **Mesure semi-additive** : La mesure peut être additionnée sur certaines dimensions uniquement. Dans notre exemple, la mesure « Stock de téléphones » est semi-additive. Le stock ne peut pas se sommer sur la dimension temps. Un stock de 10 le lundi, 8 le mardi, 8 le mercredi, 8 le jeudi, 7 le vendredi, 5 le samedi et 5 le dimanche ne signifie pas que sur la semaine le stock est de « 10 + 8 + 8 + 8 + 7 + 5 + 5 + 5 ». Par contre le stock peut être additionné sur la dimension « point de vente » qui donne bien le stock de téléphones sur l'ensemble des points de vente (ou de regroupement de points de vente selon la hiérarchie de la dimension).
- **Mesure non additive** : La mesure ne peut être additionnée sur aucune dimension. Dans notre exemple, l'indicateur « nombre de mois d'engagement du forfait » est

non additif. Il est possible de réaliser la moyenne du nombre de mois d'engagement, un écart-type pour connaitre la dispersion de la durée des engagements, etc. Mais il n'est pas possible de réaliser de somme. Vendre 3 forfaits de 12 mois d'engagement ne signifie pas qu'il y ait 36 mois d'engagement.

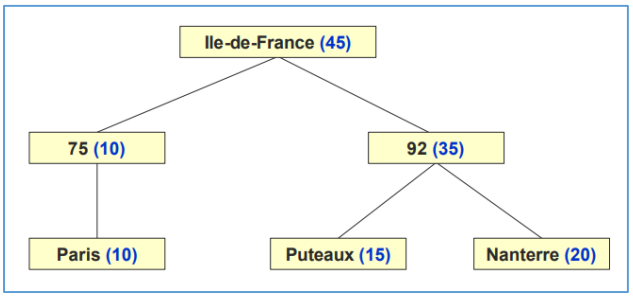


*Figure 26 : Exemple d'additivité sur une dimension et hiérarchie de localisation, source http://www.sas.com/offices/europe/france/services/support/articles/SAS_Forum_Tech_OLAP.pdf*

Le cube OLAP peut être implémenté soit par MOLAP ou ROLAP. Il est également possible de mixer les deux avec un HOLAP.

#### *5.2.9.2 MOLAP*

Le MOLAP (Multidimensional OLAP) est une implémentation multidimensionnelle du cube. Les données sont stockées dans une base de données multidimensionnelle. L'ensemble des combinaisons des dimensions sont pré-calculées dans le cube. Le temps de réponse pour une consultation du cube est donc instantané. Le langage d'interrogation du cube est le MDX (Multidimensional Expressions).

#### *5.2.9.3 ROLAP*

Le ROLAP (Relational OLAP) est une implémentation relationnelle du cube. C'est donc un système de base de données « classique » avec des relations entre tables. Les données ne sont donc pas pré-calculées. Elles seront à calculer à chaque interrogation de la base de données avec le langage SQL (Structured Query Language).

Une des principales modélisations est le modèle en étoile (cf. Figure 27, extrait de [19]). Il y a deux types de tables qui sont la table de FAIT et les tables de DIMENSION.

La table de fait, comme son nom l'indique, stocke les informations relatives au fait qui sont :

- Les indicateurs (montant des ventes et quantité vendue dans l'exemple ci-dessous) ;

- Les clés des tables de dimension. Si on raisonne en concepteur de bases de données, alors ces clés seront des clés étrangères liées aux différentes tables de DIMENSION.

Ainsi un fait est identifié de façon unique par l'ensemble des ID des tables de dimension.

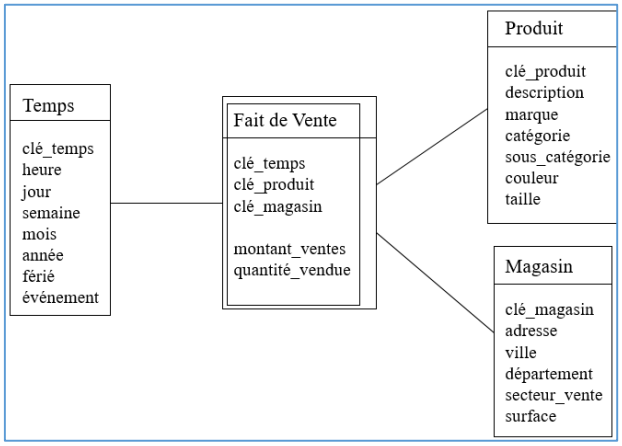


*Figure 27 : Modèle en étoile d'un fait vente, extrait de E. Metais, «Encyclopedia Universalis, chapitre "Systèmes d'aide à la décision et entrepôts de données" ISBN 978-2-85229-337-3,» 2010*

Le modèle en flocon est un modèle en étoile dont les tables de dimension sont normalisées afin d'éviter la redondance d'information (cf. Figure 28, extrait de [19]). Dans l'exemple, la sous-catégorie d'un produit se déduit de la clé du produit. Dans la procédure de normalisation, on crée donc une nouvelle table avec un nouvel ID.

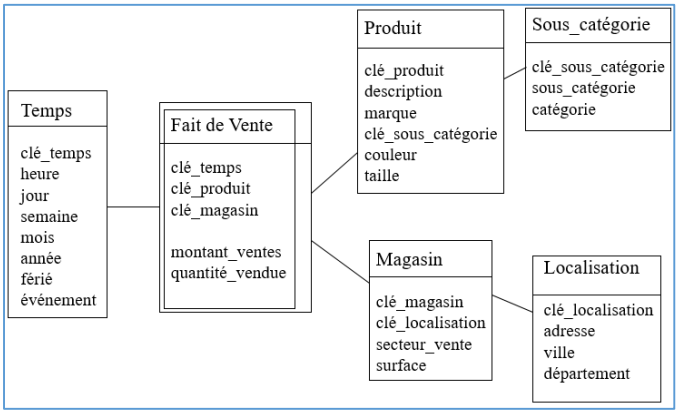


*Figure 28 : Modèle en constellation d'un fait vente, extrait de E. Metais, «Encyclopedia Universalis, chapitre "Systèmes d'aide à la décision et entrepôts de données" ISBN 978-2-85229-337-3,» 2010*

Il existe également le modèle en constellation, qui est un modèle en étoile ou flocon dont une ou plusieurs tables de dimension sont partagées par plusieurs tables de fait. Cela peut être le cas pour la dimension TEMPS qui se partage aisément avec plusieurs tables de faits.

#### *5.2.9.4 HOLAP*

Le HOLAP (Hybrid OLAP) est une implémentation double en MOLAP et ROLAP. Il cumule donc l'avantage des deux :

- Le MOLAP pour obtenir des réponses instantanées sur des agrégats de dimension ;
- Le ROLAP pour obtenir un niveau de détail plus fin soit non agrégé.

#### *5.2.9.5 Le type de fait*

Un fait peut être de trois types différents selon la manière dont on désire le mesurer.

Le type **transaction** décrit la transaction, ou l'événement, ou une action. Cela peut être l'action de vente d'un téléphone, une température à un instant donné, une entrée dans un cinéma, etc. Il est donc important que cette mesure soit horodatée (heure ou minute ou seconde ou microseconde ou plus fin selon les besoins). Dans la table de fait, il y aura donc une ligne par événement.

Le type **instantané périodique** décrit l'activité ou la performance sur une période de temps répétée régulièrement. Cela peut être le nombre de téléphones vendus par jour, le nombre d'entrée au cinéma par semaine, le temps d'ensoleillement sur la journée, etc. L'horodatage est généralement la fin de période. Dans la table de fait, il y aura donc une ligne par période de temps.

Le type **instantané récapitulatif** décrit des processus qui ont un début et une fin définis mais pas nécessairement en terme de date. Il est adapté à la mesure d'un flux d'activité ou de travail. Cela peut être la conception d'un téléphone, le traitement d'une quelconque demande, etc. L'horodatage sera souvent multiple afin de connaître les dates de passage à chaque étape du processus. Dans la table de fait, il y aura une ligne par activité mise à jour tout au long de son processus.

# 6 MIGRATION DE LA CHAÎNE DECISIONNELLE DU CALCUL DU TAUX D'USURE

La chaîne décisionnelle du calcul du taux d'usure est partiellement modifiée. Il faut définir un nouveau système d'information (ROSTAM) de stockage des données relatives aux données de l'usure. En raison de ce nouveau modèle de données, il est nécessaire de faire évoluer le traitement SAS du calcul des taux d'usure. Etant le chef de projet en charge de la partie SAS, je suis en relation avec les équipes responsables de la migration du système d'information, et également avec l'équipe métier responsable de la diffusion officielle de ces chiffres.

## 6.1 CAHIER DES CHARGES

Avant mon projet, deux chaines de traitement de la DGS avaient déjà été migrées vers ROSTAM (je n'ai pas participé à ces migrations). Deux cahiers des charges de migration de système d'information avaient donc déjà été écrits. La rédaction du cahier des charges de la migration du système d'information des données relatives à l'usure s'est alors grandement inspirée de l'existant. Le but du projet ROSTAM étant de créer un système d'information mutualisé pour les chaînes de production statistiques de la DGS, beaucoup de tables SQL et fonctionnalités sont donc uniquement à utiliser et non à créer. Le nouveau système mis en place doit respecter les contraintes ci-dessous :

- Collecter les données ;
- Pouvoir mettre à jour les données de collecte avec de nouvelles remises ;
- Conserver l'historique des remises ;
- Avoir un suivi des collectes ;
- Pouvoir requêter les données dans ROSTAM.

Cette partie est réalisée principalement par les équipes de MOA/MOE ROSTAM. Je participe au choix de la conception du nouveau modèle de données de l'usure en tant qu'utilisateur avancé de la chaîne de traitement.

Pour l'adaptation SAS du calcul du taux d'usure, je suis le responsable de l'ensemble du travail. Il s'agit de répondre aux contraintes ci-dessous :

- Obtenir des résultats identiques à l'ancien système ;
- Conserver le découpage des quatre étapes principales (chargement, contrôle, écrêtage, calcul) et le format des résultats ;
- Pouvoir envoyer les résultats à ROSTAM ;

- Conserver un temps d'exécution de traitement inférieur à 30 minutes.

La contrainte fondamentale est d'avoir les mêmes fonctionnalités et résultats qu'avec l'ancienne version.

La migration du système d'information n'amène aucune évolution métier ou fonctionnelle. Il faut donc faire évoluer les parties de traitement communicantes avec la base de données externes. Il est également nécessaire de mettre en place la transmission des données résultats à l'application ROSTAM.

Lors de la rédaction du cahier des charges, le métier responsable de la production de l'usure a indiqué que le mode de lancement de la chaîne du taux d'usure ne convenait pas. Il était réalisé via une IHM JAVA, où il s'avérait nécessaire de lancer successivement les étapes suivantes : étape de chargement, étape de contrôle, étape d'écrêtage, et enfin l'étape de calcul. Cela faisait donc quatre lancements à réaliser pour une seule production. Pour chaque étape (sauf la première du chargement), il était par ailleurs nécessaire de préciser sur quelle étape précédente se baser. Par exemple, il était possible de réaliser plusieurs chargements et de préciser que le contrôle se basait sur le chargement X. La chaîne était majoritairement lancée entièrement et donc ce mode ne convenait pas. J'ai proposé de réaliser une IHM de lancement similaire en SAS et d'ajouter la possibilité d'exécuter par défaut la totalité de la chaîne. Le lancement de chaîne sera ainsi plus rapide et la maintenance plus aisée. Nous avons plus de compétence en SAS que JAVA.

Il a été défini que je réaliserai les mises à jour des spécifications techniques, le développement, la recette technique et la mise en production de la partie SAS. Le métier en charge de la production de l'usure réalisera, après mes tests techniques, une recette fonctionnelle. Les évolutions et les acteurs sont donc clairement définis. Vu que j'utilise les données du nouveau système, je dois attendre que les évolutions en bases de données soient réalisées avant de pouvoir débuter mes développements et tests.

## 6.2 PLANNING DU PROJET

Le planning initial du projet de l'adaptation de l'usure (cf. Figure 29) s'étalonne du 19/09/2016 au 27/04/20017.

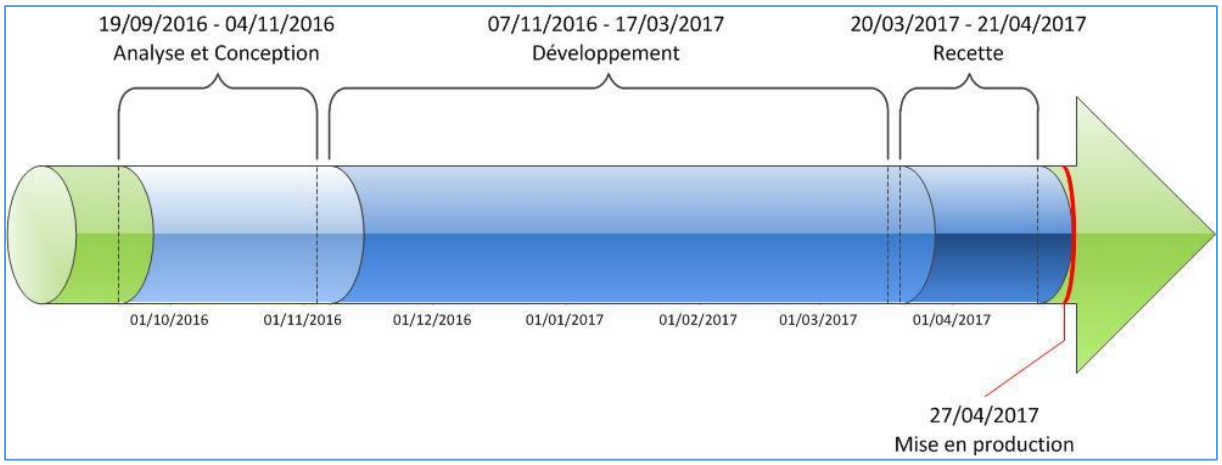


*Figure 29 : Planning initial de mise en œuvre*

La date de mise en production a été décalée au 15/06/2017, principalement à cause d'un problème de collecte de données dans ROSTAM. En effet, lors de mes développements, j'ai observé un important écart d'observations par rapport à l'ancien système. L'équipe gérant cette partie a donc dû réaliser des correctifs, pour obtenir la conformité des données chargées.

## 6.3 ANALYSE

Pour la migration partielle de la chaîne décisionnelle de l'usure, il y a trois analyses distinctes :

- L'analyse de l'intégration du fichier de collecte dans ROSTAM ;
- L'analyse du nouveau modèle de données ROSTAM pour les données de l'usure ;
- L'analyse des évolutions SAS de la chaîne de calcul des taux d'usure.

N'ayant pas participé à celle de l'intégration du fichier de collecte dans ROSTAM, j'ai rédigé un résumé en annexe à titre informatif.

### 6.3.1 Analyse du modèle de données ROSTAM

Le prérequis de base est d'utiliser une partie du modèle de données ROSTAM existant. En effet, la DGS souhaite obtenir un système d'information plus unifié. Ainsi les chaînes statistiques de production de la DGS utiliseront toutes la base de données ROSTAM

La solution proposée respecte donc ce besoin (cf. Figure 30). Partie gauche l'existant, où les chaînes de traitements (USURE, BSI, MIR, BLS…) utilisent des bases de données différentes (SMF, DTOM…). Partie droite la cible où la base unique est ROSTAM.

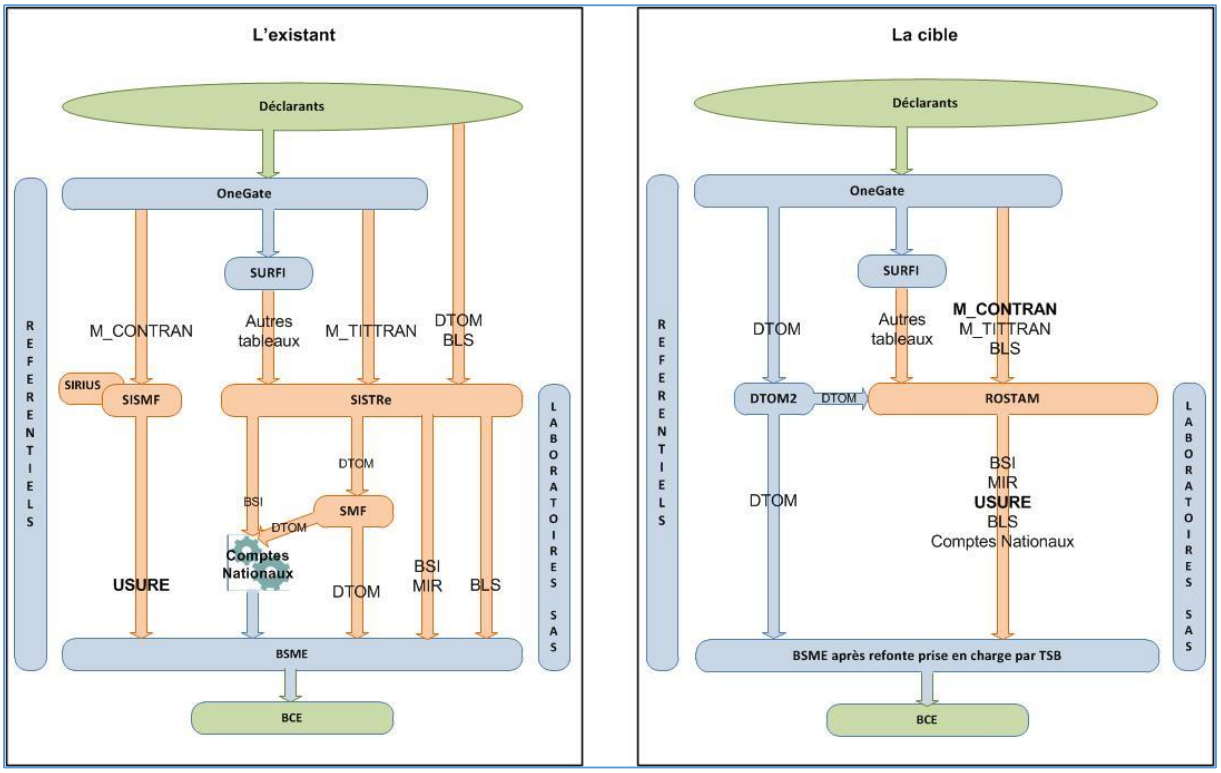


*Figure 30 : Objectif de mutualisation de la chaîne décisionnelle de la DGS*

En collaboration avec la MOE en charge de la migration du système d'information de l'usure, nous avons analysé puis proposé un modèle de données répondant aux attentes métiers. Nous étions en accord sur l'utilisation du modèle en étoile précédemment mis en place pour d'autres chaînes de traitement de la DGS. Nous profitons ainsi des tables de dimension déjà existantes. J'ai néanmoins eu, avec la MOE un point de désaccord sur l'utilisation de la table de fait existante.

Je préconise alors de créer une nouvelle table de fait pour stocker les informations de crédit de la collecte de l'usure. La majorité des données explicatives (i.e. dimension), le fait, la période du fait et les indicateurs sont différents des autres collectes de données. Il n'y a, selon moi, aucune raison de stocker les résultats dans la même table de fait que les autres collectes. De plus la structure de la table de fait existante ne me convient pas. Elle contient des champs caractères de différentes tables de dimension et non uniquement les identifiants numériques des dimensions. Malgré ces arguments, la MOE a néanmoins utilisé la même table de fait en opposant un coût de mise en œuvre moins élevé. Les choix

« informatiques » sont réalisés par l'OI et non la DGS. Je n'ai donc pas pu défendre davantage mes idées. Ces choix seront discutés dans le chapitre « 6.8 Discussion des choix ».

Le nom des variables ajoutées dans la table des faits sont identiques à l'ancien système (SISMF) ce qui permet de minimiser l'impact des traitements.

### 6.3.2 Analyse des évolutions SAS

L'analyse des impacts SAS est simplifiée car aucune modification fonctionnelle n'est réalisée. Les étapes à modifier/créer sont alors celles d'extraction des données, d'envoi des données à ROSTAM et de la nouvelle IHM de lancement de chaîne.

#### *6.3.2.1 Les extractions de données de ROSTAM*

J'ai échangé avec la MOE pour définir la meilleure méthode (i.e. la plus rapide en temps d'exécution) pour extraire les données. Afin de limiter les impacts sur la chaîne SAS, je conserve le même format des tables résultats d'extraction. Au besoin, je réalise alors des opérations de transformation pour obtenir un format identique. Ainsi, la suite de la chaîne de traitement n'est pas impactée.

Lors de cette analyse (mais également lors de la conception et de l'implémentation), j'ai demandé à connaître les optimisations positionnées sur les tables ROSTAM (principalement index et partition). Je n'ai jamais obtenu cette information. Ayant uniquement un accès via SAS à la base, je ne peux donc pas connaître cette information. Cela est bien entendu pénalisant pour définir les méthodes d'extraction. La MOE m'a transmis que si les temps d'extractions étaient trop longs, elle en analyserait les raisons et optimiserait le modèle.

SAS n'extrait pas les données directement des tables ROSTAM de production. ROSTAM met en place des vues qui sont des « copies conformes » des tables de production. C'est la politique menée à la Banque de France pour l'accès aux bases de données externes.

#### *6.3.2.2 L'identification de la catégorie d'usure de crédit*

J'ai découvert que les catégories du taux d'usure sont identifiées avec des caractéristiques des lignes de crédit (durée du crédit, montant du crédit, etc.) écrites directement dans les

programmes SAS. Cette solution ne permet pas d'obtenir un suivi des évolutions des catégories (par décret de la Loi). La seule chose que l'on puisse réaliser dans ce cadre est, soit de versionner les programmes, soit de conserver les anciennes versions avec des commentaires. Dans ces deux cas, obtenir les évolutions des catégories sur une période doit être fait en recherche textuelle. Pour exécuter une chaîne sur une échéance passée, il faut donc retrouver le bon code de cette échéance. A mon avis, ce n'est pas la bonne méthode. J'ai alors proposé de réaliser cette identification dans une table de paramètre avec des périodes de validité. Il sera ainsi possible de tracer les évolutions mais également de pouvoir lancer une chaîne sur n'importe quelle échéance sans avoir à modifier le code SAS. Cette proposition a été validée par le métier car elle n'est pas coûteuse à mettre en place et offre un réel intérêt de maintenance et de suivi. Dans l'idéal, il aurait été préférable de définir une nouvelle table de dimension dans le modèle en étoile ROSTAM. Mais cette solution a été refusée par la MOE.

#### *6.3.2.3 Nouveau mode de lancement de la chaîne SAS de l'usure*

Le lancement de la chaîne SAS sur l'ancien système se faisait via une IHM JAVA. Afin d'éviter d'être dépendant d'une autre technologie, mon analyse propose une solution complète SAS avec un lancement de la chaîne par une procédure stockée SAS. Cette dernière reprendra exactement les mêmes paramètres et fonctionnalités que l'IHM JAVA. Au niveau des procédures stockées de SAS, une gestion « d'invité » permet rapidement de créer une IHM. Le coût de création est finalement minime et la maintenance est simple. Ce choix a donc été validé. Sur l'ancien système, le lancement de la chaîne de l'usure se fait obligatoirement sur une unique échéance. J'ai proposé de pouvoir saisir de multiples échéances. Cette proposition a été rejetée par le métier car jugée non-nécessaire.

#### *6.3.2.4 Envoi des données résultats à ROSTAM*

Une fois les données de l'usure validée par le métier (relance effectuée auprès des remettants), les données sont transmises à ROSTAM pour diffusion. Pour stocker les données résultats dans la base de données ROSTAM, je conseille d'utiliser la possibilité de SAS à alimenter des bases de données externes. En effet, SAS est un ETL. Il a donc la capacité à charger des données en base de données externe. En créant un compte technique sur une

table de la base ROSTAM en « lecture-écriture », SAS peut alimenter directement la table de résultat à l'aide du compte. La MOE a une nouvelle fois refusé ma proposition au profit de l'EAI. La raison principale est que la MOE souhaite contrôler les données avant chargement. Toutefois cela aurait été possible, en donnant un accès en écriture dans une table « temporaire » de ROSTAM. Une fois les données chargées dans cette table, la MOE pouvait contrôler les données avant de les charger dans la table finale par exemple avec la mise en place d'un trigger. Ce choix n'a pourtant pas été décidé. Afin d'éviter aux utilisateurs des manipulations successives, je propose de mettre en place une procédure stockée SAS qui générera le ficher en entrée de ligne EAI. Une fois le fichier déposé par la procédure stockée, EAI effectuera le transfert et ROSTAM intégrera les données.

À l'issu de cette analyse, j'ai estimé le coût des évolutions SAS à 105 jours :

- Analyse 5j ;
- Conception- Spécifications techniques 15j ;
- Implémentation 70j ;
- Recette 15j ;

### 6.3.3 Chaîne décisionnelle proposée

L'analyse permet de proposer la chaîne décisionnelle suivante (cf. Figure 31), dont les étapes 2, 3 et 4 sont à créer/modifier :

1- Dépôt dans OneGate des données de la collecte de l'usure par les banques de crédit ;
2- Envoi à ROSTAM du fichier XML valide de la collecte de l'usure ;
3- Traitement en SAS du calcul des taux d'usure ;
4- Envoi des données résultats à ROSTAM via EAI.

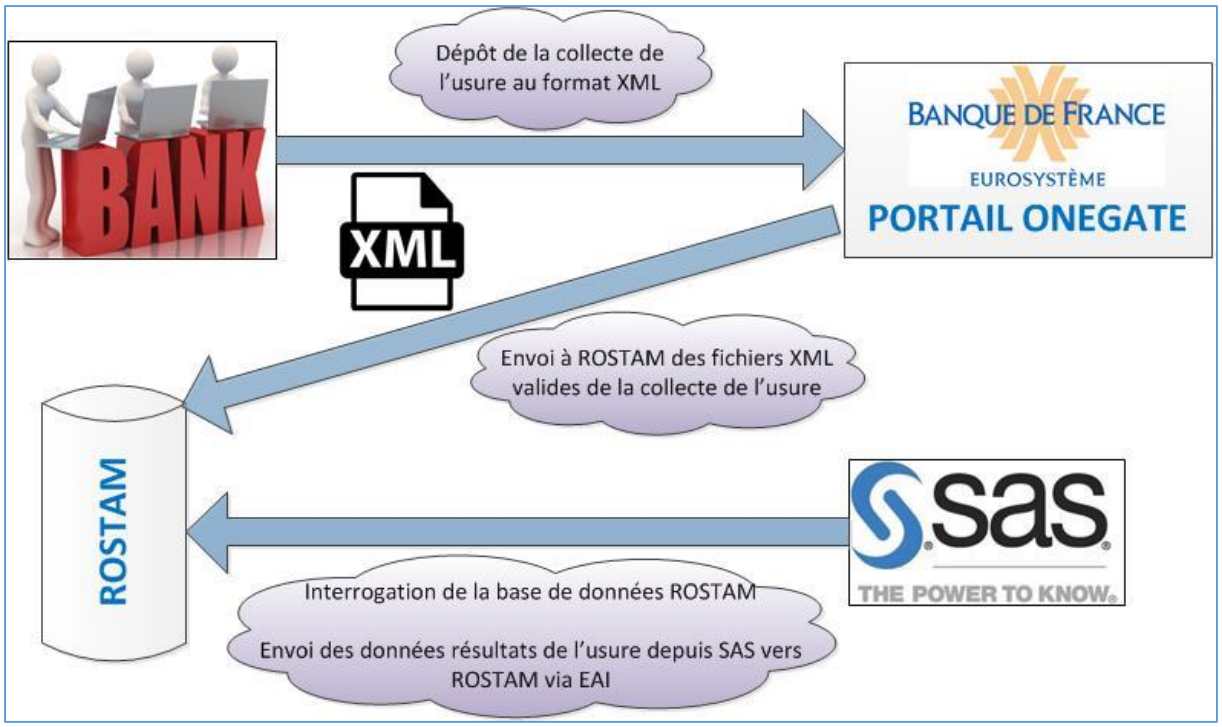


*Figure 31 : Solution proposée pour la migration de la chaîne décisionnelle de l'usure*

## 6.4 CONCEPTION

Je suis en charge de la conception complète des évolutions SAS. Ce sont principalement les étapes d'extraction et d'envoi des données résultats qui ont nécessité une conception approfondie. Les autres étapes ont nécessité des adaptations mineures. Je détaillerai donc uniquement la conception de ces deux parties.

### 6.4.1 Conception de l'extraction des données de collecte

Cette étape extrait les lignes de crédit de M_CONTRAN par échéance.

#### *6.4.1.1 Extraction des données de collecte*

L'extraction des données de collecte nécessite trois informations. L'identifiant de la collecte, l'échéance et la liste des remises. Avec ces informations, je peux extraire les données souhaitées de la table des faits V_FCT_FAIT.

Dans la base de données ROSTAM, chaque collecte de données est identifiée par un numéro unique. Il est stocké dans la table de dimension V_DIM_SOURCE (cf. Figure 32). L'usure (source = MCO) correspond à l'identifiant 216.

| | SRC_Id | SOURCE |
|---|---|---|
| 1 | 1 | Surfi |
| 2 | 2 | MTitran |
| 3 | 5 | Référentiel_TAXO_Tableaux |
| 4 | 6 | Référentiel_TAXO_Validite |
| 5 | 7 | Référentiel_TAXO_CombinaisonTCC |
| 30 | 216 | MCO |

*Figure 32 : Extrait de la table V_DIM_SOURCE*

De façon analogue, chaque échéance a un identifiant unique défini dans la table de dimension V_DIM_ECHEANCE (cf. Figure 33). Cette table ne contient pas de variable date mais deux variables numériques (mois et année). L'identifiant de l'échéance est alors la « concaténation » de ces deux variables. Ce choix sera discuté dans le chapitre « 6.8.2 La mise en œuvre du modèle étoile de la base de données ROSTAM ». Dans ma conception, j'opte pour extraire l'identifiant de l'échéance et le stocker dans une macro variable SAS. Ainsi je n'ai pas besoin dans mes requêtes SQL de faire appel à la table V_DIM_ECHEANCE. Il n'y a donc plus besoin de réaliser une jointure avec cette table. Il suffira uniquement de position une clause « where ». Cela optimise le traitement.

| | ECH_Id | MOIS | ANNEE |
|---|---|---|---|
| 1 | 11978 | 1 | 1978 |
| 2 | 11979 | 1 | 1979 |
| 3 | 11980 | 1 | 1980 |
| 4 | 11981 | 1 | 1981 |
| 5 | 11982 | 1 | 1982 |
| 6 | 11983 | 1 | 1983 |
| 7 | 11984 | 1 | 1984 |

*Figure 33 : Extrait de la table de dimension V_DIM_ECHEANCE*

La table V_DIM_TABLEAU (cf. Figure 34) stocke une ligne par remise d'un CIB. Pour chacune de ces lignes, un identifiant unique « TAB_ID » est incrémenté d'une unité. Par exemple, pour le tableau de l'usure un CIB remettra plusieurs lignes qui correspondront aux crédits accordés sur l'échéance. La table V_DIM_TABLEAU stockera alors une unique ligne. Il ne faut donc pas considérer, comme son nom le laisse supposer, cette table comme une dimension mais comme une table de fait. Le « fait » est qu'un CIB a accordé au moins un crédit sur l'échéance.

| | TAB_Id | ECHEANCE_TAB | CIB | VERSION | DATE_REMISE_OG | STATUT_CTRL | SRC_Id |
|---|---|---|---|---|---|---|---|
| 1 | 2164830 | 30APR2017 | 10228 | 1 | 23MAY2017:17:47:37.2... | INF_CH | 216 |
| 2 | 2164758 | 30APR2017 | 10548 | 1 | 17MAY2017:11:16:47.6... | INF_CH | 216 |
| 3 | 2164782 | 30APR2017 | 11600 | 1 | 18MAY2017:15:57:44.1... | INF_CH | 216 |
| 4 | 2164914 | 30APR2017 | 12549 | 1 | 29MAY2017:17:58:56.9... | INF_CH | 216 |
| 5 | 2164783 | 30APR2017 | 13070 | 1 | 18MAY2017:16:10:45.2... | INF_CH | 216 |
| 6 | 2164857 | 30APR2017 | 13106 | 1 | 24MAY2017:15:54:21.2... | INF_CH | 216 |
| 7 | 2164775 | 30APR2017 | 13150 | 1 | 18MAY2017:11:03:38.1... | INF_CH | 216 |
| 8 | 2164885 | 30APR2017 | 14406 | 1 | 24MAY2017:16:23:52.7... | INF_CH | 216 |

*Figure 34 : Extrait de la table V_DIM_TABLEAU*

La variable booléenne VERSION permet de savoir si la ligne est une remise active (i.e. la dernière du remettant). En effet, un remettant (CIB) peut déposer plusieurs fois un fichier pour une même échéance afin de corriger des erreurs. Sa remise est toujours en mode « annulée/remplacée ». Dans ce cas, pour la dernière remise, la variable VERSION sera égale à 1 et pour les autres elle vaudra 0. C'est le mode pris pour l'ensemble des tables ROSTAM pour lesquelles une notion de remise est présente. On remarque que cette variable est caractère (un « A » au niveau de la colonne). Il aurait été plus judicieux de la mettre en format numérique d'un bit.

En réalisant un filtre sur l'échéance (ECHEANCE_TAB) et sur la source (SRC_ID), on extrait les données d'une collecte sur une échéance précise.

La table V_FCT_FAIT (cf. Figure 35) stocke les données des collectes de la DGS utiles pour réaliser ses chaînes statistiques. C'est « la table de fait » du modèle en étoile de la base ROSTAM. Par contre, selon les collectes, ce ne sont pas, ni les mêmes périodes, ni le même niveau de granularité, ni les mêmes indicateurs. Il est donc difficile d'identifier clairement le fait que stocke cette table. Nous pouvons dire qu'elle stocke le fait de déposer une remise sur une échéance donnée. Ce choix de stockage unique sera discuté dans le chapitre « 6.8.2 La mise en œuvre du modèle étoile de la base de données ROSTAM ». Pour le traitement de l'usure, « le fait » est d'accorder un crédit. Il y a donc une ligne par crédit. Les indicateurs remontés sont, par exemple, le revenu annuel de la personne (ou foyer) ayant contracté le crédit et le montant du crédit.

| INS_FI | MT_CRDT | DUREE_IN | TESE | TEG | MT_REMBRST | REVENU_ANN |
|---|---|---|---|---|---|---|
| 310 | 36000 | 61 | 35000 | 36800 | 680 | 48914 |
| 310 | 10810 | 49 | 59000 | 63600 | 271 | 18564 |
| 650 | 13001 | 204 | 21000 | 28700 | 927 | 33091 |
| 680 | 62783 | 324 | 0 | 2700 | 523 | 22842 |
| 310 | 3500 | 37 | 55000 | 60300 | 108 | 26808 |
| 650 | 50356 | 324 | 25000 | 30300 | 470 | 34972 |
| 650 | 5734 | 204 | 21000 | 28900 | 571 | 29856 |
| 680 | 42168 | 240 | 29500 | 39200 | 241 | 9445 |

*Figure 35 : Extrait de la table V_FCT_FAIT pour les données de l'usure*

La table V_FCT_FAIT contient l'identifiant « TAB_ID » qui permet de réaliser la jointure avec la table V_DIM_TABLEAU. Il faut, en plus de cette jointure, filtrer sur l'échéance (ECH_ID) et sur la version active (VERSION). Le cumul de ces restrictions permet d'obtenir l'ensemble des lignes de crédit pour le calcul du taux d'usure sur une échéance donnée.

Nous verrons dans le chapitre « 6.5 Implémentation » que la jointure entre la table de fait et V_DIM_TABLEAU est inutile. Au moment de la conception, je n'en n'avais pas connaissance.

#### *6.4.1.2 Identification de la catégorie d'usure du crédit*

Comme dit précédemment dans le chapitre « 6.3 Analyse », l'identification des catégories de crédit de l'usure étant réalisée en ligne de programmation, j'ai décidé de la réaliser dans une table SAS de paramètres (cf. Figure 45) afin d'améliorer le suivi et le lancement de période passée. Cette table SAS contiendra l'ensemble des caractéristiques qui identifie les catégories de crédit de l'usure avec une notion de date de validité. Une catégorie, à un instant donné, est définie par une unique ligne dans la table. Chaque caractéristique d'un crédit est stockée dans une colonne. Si le contenu de la table doit évoluer (par un décret de Loi), un programme ad-hoc sera alors développé pour la mettre à jour. Il n'a pas été choisi de réaliser un programme de mise à jour automatique au vu de la faible fréquence de modification.

| | DT_DEB_VAL | DT_FIN_VAL | DUREE_INF | DUREE_SUP | MONTANT_SUP | EMPRUNTEURS | CODE_CATEG | LIBELLE |
|---|---|---|---|---|---|---|---|---|
| 1 | 01JUL2010 | 30SEP2012 | 0 | 24 | 76224 | PMSAM | AI21 | AUTRES PRETS... |
| 2 | 01OCT2012 | 31JUL2016 | 0 | 24 | 100000000 | PMSAM | AI21 | AUTRES PRETS... |
| 3 | 01AUG2016 | 31DEC9999 | 0 | 24 | 100000000 | PMSAM | AI21 | AUTRES PRETS... |
| 4 | 01JUL2010 | 31JUL2016 | 0 | 24 | 76224 | SNF | AI22 | AUTRES PRETS... |
| 5 | 01AUG2016 | 31JAN2017 | 0 | 24 | 76224 | SNF | AI22 | AUTRES PRETS... |
| 6 | 30APR2017 | 31DEC9999 | 0 | 24 | . | SNF | AI22 | AUTRES PRETS... |

*Figure 36 : Extrait de la table SAS des paramètres des catégories de l'usure*

Cette table me permettra de créer une macro variable SAS contenant l'ensemble des caractéristiques des crédits. Cette macro variable sera de type : DUREE_INF ≥ xxx and DUREE_SUP ≤ xxx and MONTANT_SUP ≤ xxx… Je pourrai ainsi tester chacune des lignes de crédit avec ces macros variables et identifier sa catégorie de crédit de l'usure.

### 6.4.2 Conception du nouveau mode de lancement de la chaîne SAS de l'usure

Le nouveau mode de lancement sera exécuté avec une procédure stockée SAS. Elle reprend les possibilités de l'ancien mode de lancement (IHM JAVA) auxquelles j'ajoute l'exécution par défaut de l'ensemble de la chaîne. Elle permet alors de lancer la chaîne, visualiser ou supprimer un ancien lancement. J'ai laissé la possibilité de lancer une étape à partir d'un prédécesseur au choix. Les utilisateurs ont ainsi, comme pour l'ancienne application, une IHM qui permet facilement de lancer la chaîne de traitement.

Les paramètres de saisi dans l'IHM sont utilisables dans les programmes. SAS crée automatiquement des macros variables avec les paramètres saisis.

### 6.4.3 Conception de l'envoi des données résultats à ROSTAM

Une fois les données résultats de l'usure validées par le métier, les données sont transmises à ROSTAM via EAI. L'EAI mis en place prend un fichier ZIP en entrée. J'ai choisi de créer ce fichier avec une procédure stockée SAS.

L'utilisateur renseigne uniquement le processus de calcul sur lequel se baser. En effet, plusieurs étapes de calcul peuvent être lancées. Il faut donc en choisir une. La mise en place de EAI est sous la responsabilité de l'OI, j'ai celle de déposer un fichier ZIP (dont le contenu est formalisé) dans un endroit précis.

### 6.4.4 Conception du modèle en étoile ROSTAM

Je ne participe pas à la conception de ce modèle en étoile. Je participe uniquement à son analyse. Mais afin de comprendre sa structure, j'ai reconstitué un extrait simplifié du modèle de données des principales tables qui sont utilisées dans les traitements SAS (cf. Figure 37).

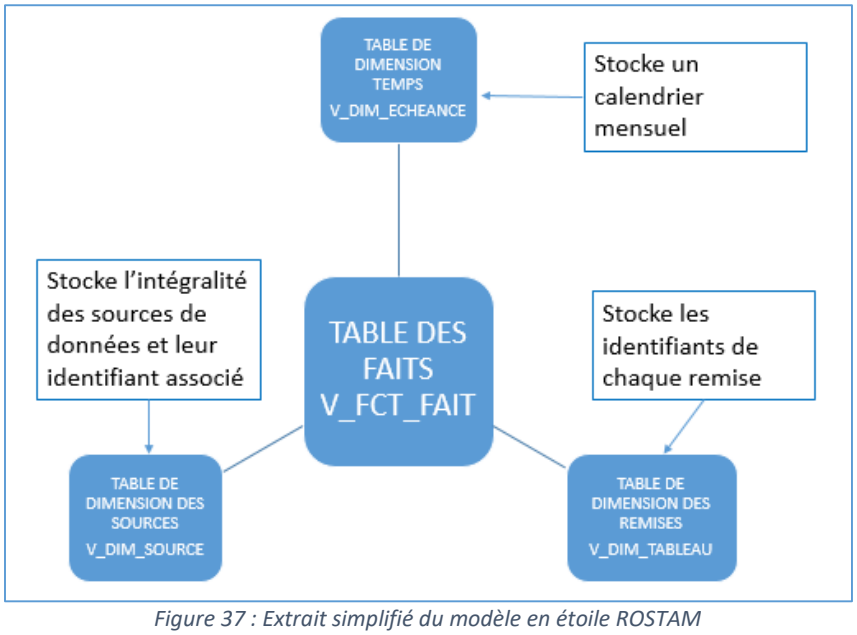


*Figure 37 : Extrait simplifié du modèle en étoile ROSTAM*

Je place également en annexe un extrait de la conception du nouveau modèle de données (cf. Figure 89) issu de l'analyse.

## 6.5 IMPLEMENTATION

J'explique l'implémentation de la partie SAS car je ne participe pas à celle de l'intégration des données dans ROSTAM.

### 6.5.1 Les différents environnements

Pour réaliser les développements, j'ai trois environnements à ma disposition :

- L'environnement de développement a une moindre puissance et une moindre volumétrie de données qui ne permet pas de valider intégralement la chaîne. J'ai donc réalisé les développements et les tests unitaires. Une fois ces derniers validés, j'ai livré mes programmes en environnement d'intégration ;
- L'environnement d'intégration contient les données identiques à la production. Une fois le cahier de test validé, j'ai livré la chaîne SAS dans l'environnement en production ;
- L'environnement de production qui, comme son nom l'indique, sert à réaliser les diverses productions.

### 6.5.2 Stockage des données SAS intermédiaires et résultats

Je conserve le fonctionnement du stockage des données SAS, car il convient parfaitement au métier. Lors de l'exécution d'une étape de la chaîne, un répertoire LINUX est créé. L'intégralité des tables SAS créée lors de cette étape y sont stockées. De plus, il y a un répertoire par étape (soit quatre en totalité) pour stocker les tables identifiées comme importantes par le métier. Sur ces répertoires un libname SAS est appliqué. Afin d'aider à la validation des résultats, il y a quatre libname stockant les tables SAS de l'ancien système et quatre autres stockant les tables SAS du nouveau système. La différenciation est un « O » (pour old) à la fin du libname pour l'ancien système (cf. Figure 38).

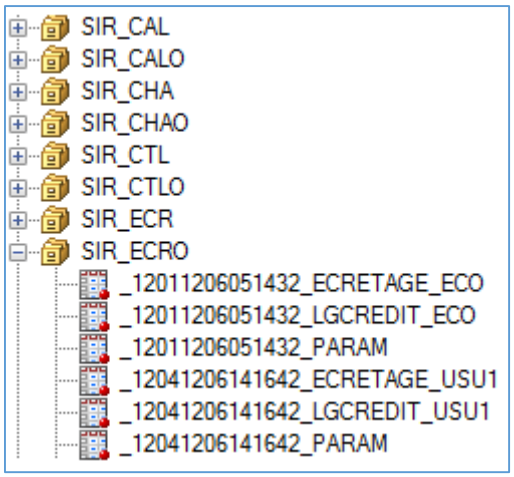


*Figure 38 : Chaîne SAS de l'usure - stockage des données SAS*

Je ne modifie pas le stockage des données mais je fais évoluer le lancement de la chaîne pour répondre aux exigences métier.

### 6.5.3 Nouveau mode de lancement de la chaîne de calcul des taux d'usure

Je réalise une procédure stockée SAS (cf. Figure 39) reprenant les possibilités de l'ancien mode de lancement (IHM JAVA) auxquelles j'ajoute l'exécution par défaut de l'ensemble de la chaîne. Elle permet alors de lancer la chaîne, visualiser ou supprimer un ancien lancement. Je laisse la possibilité de lancer une étape à partir d'un prédécesseur au choix. Les utilisateurs ont ainsi, comme pour l'ancienne application, une IHM qui permet facilement de lancer la chaîne de traitement.


*Figure 39 : Procédure stockée de calcul du taux d'usure, les paramètres de lancement*

Lors de la réalisation de l'adaptation du taux d'usure, je réponds à un nouveau besoin du métier d'obtenir un espace de simulation. L'objectif est d'étudier l'impact de modification de

catégories, de contrôles… sur les données de production mais sans modifier les programmes de l'usure. J'ai alors créé une arborescence similaire à la production pour y stocker les mêmes programmes, tables de référentiels et les résultats. La production n'est ainsi nullement impactée par ces tests. Il y a alors un paramètre « instance » qui permet de choisir sur quel environnement on veut exécuter le traitement. Par défaut le paramètre est « production ».

Il existe de multiples contrôles des données dans la chaîne dont certains peuvent être paramétrés lors du lancement. J'ai donc repris ces paramètres et laissé la possibilité aux métiers de mettre les valeurs choisies (cf. Figure 40). Ceci est le cas dans l'ancienne version.

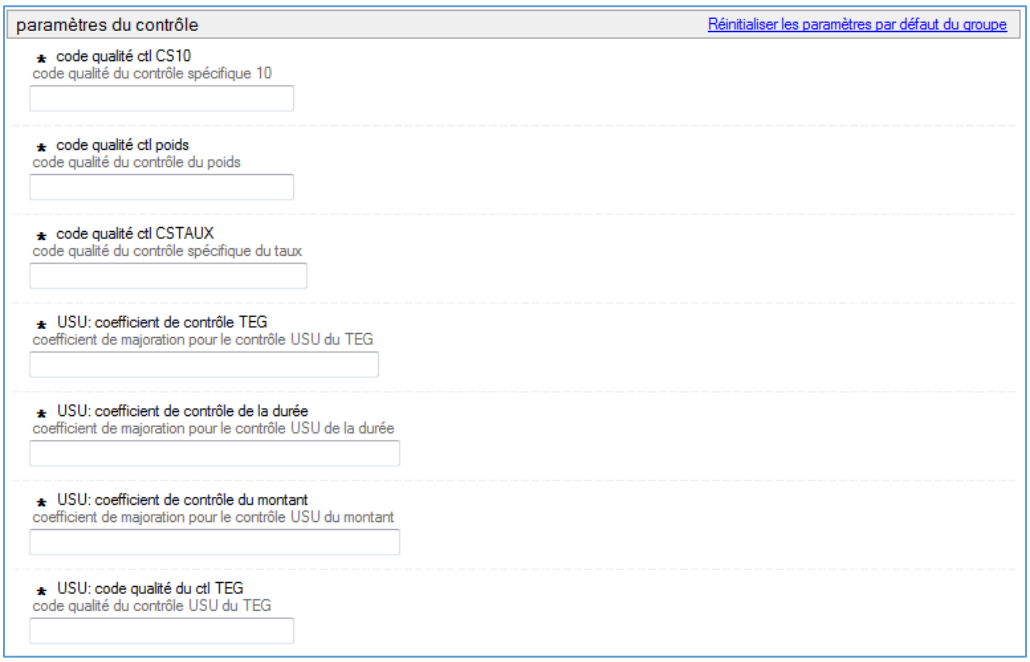


*Figure 40 : Procédure stockée de calcul du taux d'usure, les paramètres de contrôles*

SAS crée automatiquement cette IHM en ajoutant des « Invites » (cf. Figure 41). Par contre, il n'y a pas le choix de la mise en page qui est « succincte ». Ces « invites » créent des macro-variables SAS que l'on peut utiliser dans le traitement. Leur nom est défini dans la colonne « Nom ». J'ai également repris le même nom de paramètre que dans l'ancien système. L'impact de ce nouveau mode de lancement sur le traitement SAS est ainsi nul.

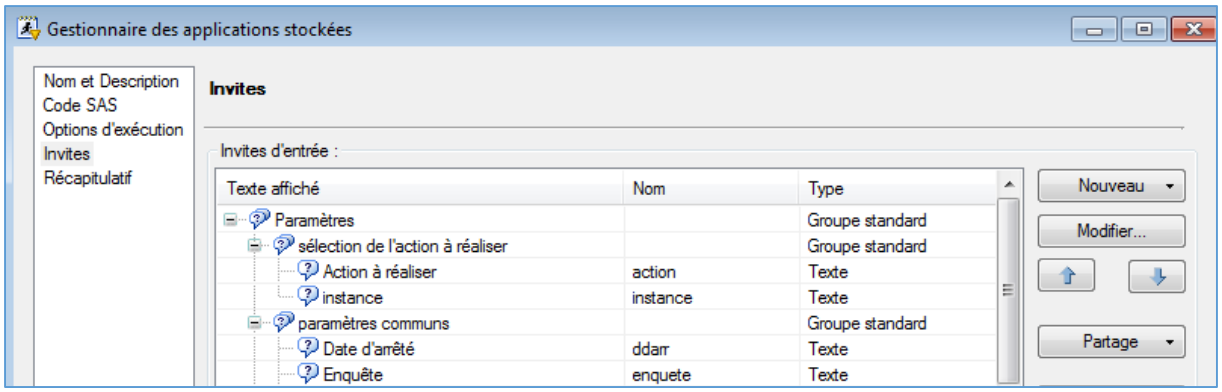


*Figure 41 : Procédure stockée de calcul du taux d'usure, création de l'IHM*

La procédure stockée permet de lancer le programme initial du calcul de l'usure (cf. Figure 42). On y retrouve le paramètre de l'instance qui spécifie si l'on désire travailler sur l'espace de simulation ou de production.

```
%MACRO lancement_instance();
%IF %UPCASE(&instance.)=PRD %THEN %DO;
        %INCLUDE '$DE2SM_GW_USURE_SAS_LIBRE_PROD/V002/prog/bao/gesdem_sil_sir2.bps.sas';
    %END;
    %ELSE %IF %UPCASE(&instance.)=SIM %THEN %DO;
        %INCLUDE "$DE2SM_GW_USURE_SAS_LIBRE_SIMU/V002/prog/bao/gesdem_sil_sir2.bps.sas";
    %END;
%MEND lancement_instance;

%lancement_instance();

%gesdem_sil_sir2(action=&action,
                 instance=&instance,
                 DDARR=&DDARR,
                 PREDECESSEUR=&PREDECESSEUR,
                 ...,
                 controle_eco_tese1=&controle_eco_tese1,
                 controle_eco_tese2=&controle_eco_tese2,
                 controle_eco_duree=&controle_eco_duree,
                 controle_eco_montant=&controle_eco_montant,
                 seuil_ECO_codqual=&seuil_ECO_codqual,
                 trd_code_supp=&trd_code_supp);
```

*Figure 42 : Code SAS de la procédure stockée de lancement du calcul du taux d'usure*

Le programme GESDEM_SIL_SIR2, lancera ensuite les différentes étapes du calcul de l'usure. Les différents paramètres sont ceux de l'IHM de lancement.

### 6.5.4 Adaptation de l'étape de chargement

Cette étape extrait les lignes de crédit de M_CONTRAN par échéance.

#### *6.5.4.1 Extraction des données de collecte*

Lors des échanges avec la MOE ROSTAM, nous avons défini que, pour extraire les données de l'usure de la table des faits V_FCT_FAIT, il était obligatoire d'extraire les remises actives V_DIM_TABLEAU. J'avais donc rédigé, dans les spécifications techniques, la nécessité d'utiliser cette table.

Comme expliqué dans le chapitre « 6.4.1.1 Extraction des données de collecte », la table V_FCT_FAIT contient l'identifiant « TAB_ID » qui permet de réaliser la jointure avec la table V_DIM_TABLEAU. Il faut, en plus de cette jointure, filtrer sur l'échéance avec la variable « ECH_ID » et sur la variable « VERSION » pour extraire uniquement la dernière remise. Le cumul de ces restrictions permet d'obtenir l'ensemble des lignes de crédit pour le calcul du taux d'usure sur une échéance donnée (cf. Figure 43).

```
PROC SQL NOPRINT;
     CREATE TABLE &lib_out..m_contran_&ech. AS
         SELECT f.section AS SCT LABEL="Identifiant de la section",
                f.code_guichet AS ID_GUI LABEL="Code guichet",
                f.fait_non_numerique AS RFLICR LABEL="Référence du crédit",
                ...
                f.ACTIVITE AS SAC LABEL="Activité",
                f.TYP_POP AS SCP LABEL="Type population"

         FROM rostam.v_fct_fait f INNER JOIN rostam.v_dim_tableau t
         ON f.tab_id = t.tab_id AND
            f.ech_id = t.ech_id
         WHERE f.ech_id       = &ech_id. AND
               f.last_version = 1 AND
               t.src_id       = 216
     ;
QUIT;
```

*Figure 43 : Extrait de code SAS-SQL de sélection des données du calcul du taux d'usure*

Remarque : les « … » dans le code SQL ne sont pas corrects en programmation. C'est uniquement dans un souci de présentation que ceci est ajouté.

Cette extraction est conforme aux spécifications techniques que j'ai écrites. Mais en analysant la structure de ces deux tables, et plus précisément celle de la table de fait, je ne voyais pas l'utilité de réaliser une jointure entre ces deux tables. En effet, par construction de ROSTAM, dès qu'une remise est correcte, une ligne est présente dans V_DIM_TABLEAU et le contenu de la remise est chargé dans V_FCT_FAIT. Le test d'existence « f.tab_id = t.tab_id » est donc inutile car toujours vrai. J'ai pu le vérifier sur plusieurs échéances et sources de données. La partie « FROM – WHERE » de la requête a donc été changée comme ci-dessous (cf. Figure 44).

```
FROM rostam.v_fct_fait
WHERE src_id = 216       and
      ech_id = &ech_id. and
      last_version=1
```

*Figure 44 : Code SQL optimisé pour l'extraction des données de la table de fait ROSTAM*

Sur l'échéance d'avril 2017, le temps d'exécution de la requête est réduit d'une minute et trente secondes à une minute et cinq secondes. Cette différence de temps d'exécution aurait augmenté en même temps que le nombre de lignes de ces deux tables.

J'ai fait part de cette optimisation pour les autres chaînes de traitement utilisant les données de la table V_FCT_FAIT. L'ensemble des chaînes de traitement, après avoir réalisé les tests, est passé avec cette nouvelle requête.

Une fois les données extraites, il est nécessaire d'identifier la catégorie du taux d'usure du crédit.

#### *6.5.4.2 Identification de la catégorie d'usure du crédit*

Comme expliqué dans le chapitre « 6.3 Analyse », l'identification des catégories de l'usure était réalisée en ligne de programmation. J'ai décidé de réaliser cette identification dans une table SAS de paramètres (cf. Figure 45) afin d'améliorer le suivi et le lancement de période passée.

| | DT_DEB_VAL | DT_FIN_VAL | DUREE_INF | DUREE_SUP | MONTANT_SUP | EMPRUNTEURS | CODE_CATEG | LIBELLE |
|---|---|---|---|---|---|---|---|---|
| 1 | 01JUL2010 | 30SEP2012 | 0 | 24 | 76224 | PMSAM | AI21 | AUTRES PRETS... |
| 2 | 01OCT2012 | 31JUL2016 | 0 | 24 | 100000000 | PMSAM | AI21 | AUTRES PRETS... |
| 3 | 01AUG2016 | 31DEC9999 | 0 | 24 | 100000000 | PMSAM | AI21 | AUTRES PRETS... |
| 4 | 01JUL2010 | 31JUL2016 | 0 | 24 | 76224 | SNF | AI22 | AUTRES PRETS... |
| 5 | 01AUG2016 | 31JAN2017 | 0 | 24 | 76224 | SNF | AI22 | AUTRES PRETS... |
| 6 | 30APR2017 | 31DEC9999 | 0 | 24 | . | SNF | AI22 | AUTRES PRETS... |

*Figure 45 : Extrait de la table SAS des paramètres des catégories de l'usure*

Lors du calcul du taux d'usure, j'extrais les caractéristiques des catégories valides sur la date échéance traitée (filtre sur les dates de début et fin de validité). Pour chacune des caractéristiques, je crée une condition SQL. L'ensemble de ces conditions est cumulé avec l'opérateur logique « ET » (AND en SQL) dans une unique variable « condition » (cf. Figure 46).

```
 /* sélection des paramètres de l'échéance traitée */
 if DT_FIN_VAL>=&dtech and DT_DEB_VAL<=&dtech ;

 /* création des conditions de définition des catégorie de l'usure */
  if M_Contrant ne " " then cond_mc=' M_contrant in ('||'"'||trim(M_contrant)||'"'||') and' ;
    else cond_mc='';
  if Usage_prt ne . then cond_usg=' Usage_prt ='||trim(Usage_prt)||' and';
    else cond_usg='';
  if code_ins ne " " then cond_ins=' code_ins in '||trim(code_ins)||' and' ;
    else cond_ins='';
  if duree_inf ne . then cond_dmin= ' duree>'||trim(duree_inf)||' and ';
     else cond_dmin='';
  if duree_sup ne . then cond_dmax= ' duree<='||trim(duree_sup)||' and ';
     else cond_dmax='';
  if montant_inf ne . then cond_mmin= ' mt_crdt>'||trim(montant_inf)||' and ';
     else cond_mmin='';
  if montant_sup ne . then cond_mmax= ' mt_crdt<='||trim(montant_sup)||' and ';
     else cond_mmax='';
 /* concaténation des conditions */
 cond_=trim(cond_mc||cond_usg||cond_ins||...||cond_rgl||cond_zone||cond_crb);
/* length-3 afin de supprimer le dernier "and" mis par défaut sur chaque condition */
condition=(substr(cond_,1,length(cond_)-3));
```

*Figure 46 : Extrait de code SAS de définition des catégories des taux d'usure*

Remarque : les « … » dans la définition de la variable « cond » ne sont pas corrects en code SAS. Je n'ai également pas noté l'ensemble des conditions « if », ceci est pour un souci de présentation, le but étant uniquement de montrer le principe de programmation.

Avec cette table de condition, je crée des macro-variables pour identifier les catégories de l'usure des lignes de crédit (cf. Figure 47).

```
/* definition des macro variable des conditions des catégories */
data _null_ ;
   set &libwork..&enquete2._conditions end=fin;
   /* macro variable de la catégorie */
   call symput('Categ_'!!left(_n_),strip(code_categ)) ;
   /* macro variable de la condition de la catégorie */
   call symput('cond_'!!left(_n_),strip(compbl(condition))) ;
   if fin then call symput('nb_cond',_n_) ;
run ;
```

*Figure 47 : Code SAS de création de macro variable SAS des conditions des catégories des taux d'usure*

Ce code génère alors des macro-variables qui définissent les catégories de l'usure. Le code ci-dessous (cf. Figure 48) montre le contenu des macro-variables categ1 et cond_1. La macro-variable Categ_1 vaut AI21, qui est le code de la catégorie. La macro-variable Cond_1 est le résultat de la concaténation de l'ensemble des conditions, soit les caractéristiques des crédits de AI21.

```
23         %put categ1 : &categ_1;
categ1 : AI21
24         %put cond_1 : &cond_1;
cond_1 : M_contrant in ("MCO4") and code_ins in ('200','210','220','230','240','260','410','440','500','510','690') and duree> 0
and duree<= 24 and mt_crdt<= 100000000 and crb not in
('15','151','153','154','155','498','608','618','648','658','668','688','808','818')
```

*Figure 48 : Exemple de macro-variable des catégories de l'usure*

Le langage SAS permet de créer automatiquement du code à l'aide du « macro langage SAS ». J'utilise cette fonctionnalité pour coder une succession de « else if », identifiant les catégories de l'usure. Une fois les macros variables créées, je les utilise dans une boucle « do » selon le nombre de conditions que contient la macro variable « nb_cond ». Les conditions sont exclusives, c'est-à-dire qu'une donnée ne peut répondre qu'à une seule catégorie de l'usure. Pour cette raison, j'utilise un « else if ».

```
data &libwork..lgcredit_&enquete2 ;
  attrib code_categ  length=$5. ;
  set &libwork..lgcredit_&enquete2 ;
    /* vérifcation si la ligne de crédit appartient à la 1ere définition de catégorie */
    if &cond_1  then do ;
      code_categ="&categ_1";
    end;
    /* sinon on teste les autres catégories. Une donnée ne peut répondre qu'à une seule condition */
    %do i=1 %to &nb_cond ;
      else if &&cond_&i then do;
        code_categ="&&categ_&i";
      end;
    %end;
run;
```

*Figure 49 : Code SAS attribuant à la ligne de crédit la catégorie de l'USURE correspondante*

SAS génère alors le code suivant (cf. Figure 50) qui contient bien l'ensemble des conditions des catégories de l'usure.

```
MPRINT(A):   if M_contrant in ("MCO4") and code_ins in ('200','210','220','230','240','260','410','440','500','510','690') and
duree> 0 and duree<= 24 and mt_crdt<= 100000000 and crb not in
('15','151','153','154','155','498','608','618','648','658','668','688','808','818') then do ;
MPRINT(A):   code_categ="AI21";
MPRINT(A):   end;
MPRINT(A):   else if M_contrant in ("MCO4") and code_ins in ('200','210','220','230','240','260','410','440','500','510','690') and
duree> 0 and duree<= 24 and mt_crdt<= 100000000 and crb not in
('15','151','153','154','155','498','608','618','648','658','668','688','808','818') then do;
MPRINT(A):   code_categ="AI21";
MPRINT(A):   end;
MPRINT(A):   else if M_contrant in ("MCO2") and code_ins in ('200','210','220','230','240','260','410','440','500','510','690') and
duree> 0 and duree<= 24 and crb not in ('15','151','153','154','155','498','608','618','648','658','668','688','808','818') then do;
MPRINT(A):   code_categ="AI22";
MPRINT(A):   end;
```

*Figure 50 : Extrait du code SAS avec la boucle sur la boucle des conditions des catégories de l'usure*

L'étape de chargement étant terminée, la suivante est l'étape de contrôle.

### 6.5.5 Adaptation de l'étape de contrôle

Ce processus contrôle les informations des lignes de crédit dans le but de détecter des anomalies de déclaration supplémentaires à celles identifiées en amont via les contrôles de conformité dans Onegate et ROSTAM. Les informations ainsi concernées sont :

- TEG pour l'usure ;
- Montant de crédit ;
- Durée initiale.

Le contrôle des données est réalisé par rapport aux valeurs moyennes par catégorie de la période précédente. Pour cette partie j'adapte uniquement l'extraction des données depuis la nouvelle base ROSTAM (cf. Figure 51). Je ne réalise aucune modification sur les contrôles. Je conserve le même format d'extraction que l'ancien système afin que le reste du programme ne soit pas impacté.

```
/* création de la macro variable de la date d'échéance précédente */
DATA _NULL_;
     FORMAT ech echp date9.;
     /* échéance en cours */
     ech = INPUT("&ddarr.", ddmmyy10.);
     /* échéance précédente soit 3 mois de mois */
     echp = INTNX('month', ech, -3, 'end');
     CALL SYMPUTX("echp",PUT(echp,date9.));
RUN;

/* extraction des résultats de l'échéance précédente */
PROC SQL;
     CREATE TABLE series_resultats_brut AS
          SELECT SCAN(nmvar,1,"_") AS code_categ LENGTH=5,
                 nmvar,
                 mtobs
          FROM ROSTAM.v_fct_fait_diff
          WHERE echeance         = "&echp."d AND
                src_id           = 224       AND
                derniere_version = 1
     ;
QUIT;
```

*Figure 51 : Code SAS d'extraction des taux d'usure de l'échéance précédente*

La première partie du programme récupère la date de l'échéance précédente (i.e. trimestre précédent). Un des paramètres de la procédure stockée de lancement de l'usure est la date d'arrêtée (échéance) nommée « ddarr ». Elle est de type « jj/mm/aaaa » soit par exemple « 31/01/2017 ». La fonction INPUT permet de créer une variable de type date contenant « 31/01/2017 ». La fonction INTNX de SAS permet de soustraire (ou additionner) des périodes à une date. En spécifiant le paramètre MONTH, j'indique la soustraction d'un nombre de mois. Le paramètre END indique que le résultat est une fin de période soit la fin du mois. Pour l'échéance de janvier 2017, la période précédente sera octobre 2016 (echp = 31OCT2016). Les données résultats de l'usure sont stockées dans la table V_FCT_FAIT_DIFF.

Remarque : Cette table ne contient pas d'identifiant de l'échéance issu de la table de dimension V_DIM_ECHEANCE, mais uniquement la date d'échéance. Le résultat est alors le suivant (cf. Figure 52) :

| | code_categ | NMVAR | Mtobs |
|---|---|---|---|
| 1 | 3A6 | 3A6_MOY_DUREE | 45 |
| 2 | 3A6 | 3A6_MOY_MONT | 4417 |
| 3 | 3A6 | 3A6_MOY_TEG | 9.9972 |
| 4 | 3A6 | 3A6_TX_USU | 13.3295 |
| 5 | AI21 | AI21_MOY_DUREE | 14 |
| 6 | AI21 | AI21_MOY_MONT | 715851 |
| 7 | AI21 | AI21_MOY_TEG | 0.9411 |
| 8 | AI21 | AI21_TX_USU | 1.2548 |

*Figure 52 : Extrait de l'extraction des indicateurs de l'usure*

- XXX_Moy_mont : La moyenne de la valeur du crédit pour la catégorie XXX accordée sur l'échéance T-1 ;
- XXX_Moy_dure : La moyenne de la durée du crédit pour la catégorie XXX accordée sur l'échéance T-1 ;
- XXX_Moy_teg : La moyenne du taux effectif du crédit pour la catégorie XXX accordée sur l'échéance T-1 ;
- XXX_Tx_usu : Le taux d'usure pour la catégorie XXX effectif sur l'échéance T-1 ;

Chaque indicateur est sur une ligne, soit quatre lignes par catégorie. Afin de ne pas impacter le code de contrôle existant, une seule ligne par catégorie est nécessaire, avec les quatre indicateurs en variable.

Je le réalise principalement avec les fonctions RETAIN et LAST de SAS (cf. Figure 53). La fonction RETAIN sauvegarde les variables spécifiées en mémoire, si on n'affecte pas de nouvelle valeur. La fonction LAST identifie le dernier tuple pour les variables spécifiées dans le BY, soit le code catégorie dans notre cas.

```
DATA &lib_out..ref_usu2_&p_dat_ech.(DROP=nmvar mtobs);
    SET &lib_out..series_resultats_brut;
    LENGTH code_categ   $5
           moy_mont      8
           moy_duree     8
           moy_teg       8
           tx_usu        8
    ;
    BY code_categ;
    /* sauvegarde des informations pour les lignes suivantes */
    RETAIN moy_mont moy_duree moy_teg tx_usu;
    /* initialisation a vide au début de chaque catégorie */
    IF FIRST.code_categ THEN DO;
        moy_duree=.;
        moy_mont=.;
        moy_teg=.;
        tx_usu=.;
    END;
    /* affectation de l'indicateur correspondant */
    IF      INDEX(UPCASE(nmvar),"MOY_DUREE") NE 0 THEN moy_duree = mtobs;
    ELSE IF INDEX(UPCASE(nmvar),"MOY_MONT")  NE 0 THEN moy_mont  = mtobs;
    ELSE IF INDEX(UPCASE(nmvar),"MOY_TEG")   NE 0 THEN moy_teg   = mtobs;
    ELSE IF INDEX(UPCASE(nmvar),"TX_USU")    NE 0 THEN tx_usu    = mtobs;
    IF LAST.code_categ THEN OUTPUT;
RUN;
```

*Figure 53 : Code SAS de transformation des indicateurs de taux d'usure sur une ligne par catégorie*

Le résultat ainsi obtenu (cf. Figure 54) est d'une ligne par catégorie avec les quatre indicateurs.

| | code_categ | moy_mont | moy_duree | moy_teg | tx_usu |
|---|---|---|---|---|---|
| 1 | 3A6 | 4417 | 45 | 9.9972 | 13.3295 |
| 2 | AI21 | 715851 | 14 | 0.9411 | 1.2548 |
| 3 | AI22 | 14720 | 7 | 1.7608 | 2.3478 |

*Figure 54 : Extrait des indicateurs du taux d'usure du trimestre précédent sur une seule ligne par catégorie*

L'exemple ci-dessous (cf. Figure 55) détaille l'apport des fonctions RETAIN et LAST. Chaque ligne renseigne un indicateur supplémentaire pour la catégorie de crédit. C'est la fonction RETAIN qui permet de le faire sinon les valeurs seraient remises à zéro à chaque ligne. La fonction LAST en combinaison avec OUTPUT conserve uniquement la dernière ligne par catégorie (BY code_categ), où l'ensemble des indicateurs est renseigné.

| | code_categ | moy_mont | moy_duree | moy_teg | tx_usu |
|---|---|---|---|---|---|
| 1 | 3A6 | . | 45 | . | . |
| 2 | 3A6 | 4417 | 45 | . | . |
| 3 | 3A6 | 4417 | 45 | 9.9972 | . |
| 4 | 3A6 | 4417 | 45 | 9.9972 | 13.3295 |
| 5 | AI21 | . | 14 | . | . |
| 6 | AI21 | 715851 | 14 | . | . |
| 7 | AI21 | 715851 | 14 | 0.9411 | . |
| 8 | AI21 | 715851 | 14 | 0.9411 | 1.2548 |

*Figure 55 : Exemple détaillant l'apport des fonctions RETAIN et LAST de SAS*

### 6.5.6 L'étape d'écrêtage

Historiquement ce processus définit le périmètre des lignes de crédit qui contribuent aux calculs de l'usure. Ce résultat est obtenu au moyen de l'écrêtage des lignes de crédit des intervalles d'extrémité des courbes de distribution calculées par catégorie/sous-critère. Ce processus d'écrêtage est conservé, mais uniquement à titre informatif. Désormais les taux officiels sont calculés sur l'ensemble des données.

À cet effet, des intervalles d'écrêtage par défaut sont paramétrés dans la table de paramétrage par Catégorie/sous-critère. Toutefois, des intervalles spécifiques pourront être passés en paramètre d'entrée du traitement et seront pris en priorité.

La table de paramètre d'écrêtage n'a pas été reprise dans ROSTAM. J'ai donc repris son contenu (cf. Figure 56). Le code de cette étape n'a pas été impacté (excepté l'extraction de la table de paramètre).

| | CTL_GCH | CTL_DRT | CODE_CATEG | CODE_SOUS_CATEG | dt_deb_val | dt_fin_val |
|---|---|---|---|---|---|---|
| 1 | 1 | 99 | AI21 | 210 | 01JUL2010 | 31DEC9999 |
| 2 | 1 | 99 | AI21 | 410 | 01JUL2010 | 31DEC9999 |
| 3 | 1 | 99 | AI21 | 440 | 01JUL2010 | 31DEC9999 |
| 4 | 1 | 99 | AI21 | 510 | 01JUL2010 | 31DEC9999 |
| 5 | 1 | 99 | DEC1 | 100 | 01JUL2010 | 31DEC9999 |
| 6 | 1 | 99 | DEC1 | 320 | 01JUL2010 | 31DEC9999 |

*Figure 56 : Table SAS de paramètre des bornes d'écrêtage*

Comme pour la table de paramètres des catégories de l'usure, il n'a pas été prévu de mise à jour automatique de cette table, au vu de la faible fréquence de la modification de ces bornes.

### 6.5.7 L'étape de calcul

Cette étape calcule les taux d'usure officiels et elle n'a pas été impactée par les évolutions. Les indicateurs publiés sont donc directement extraits des tables SAS (cf. Figure 57) :

- Moy_mont : La moyenne de la valeur du crédit accordé sur l'échéance T ;
- Moy_dure : La moyenne de la durée du crédit accordé sur l'échéance T ;
- Moy_teg : La moyenne du taux effectif du crédit accordé sur l'échéance T ;
- Tx_usu : Le taux d'usure effectif sur l'échéance T+1.

| | code_categ | moy_mont | moy_duree | moy_teg | tx_usu |
|---|---|---|---|---|---|
| 1 | 3A6 | 4410 | 45 | 9.8429 | 13.123891904 |
| 6 | IA3 | 346 | 13 | 15.4523 | 20.603103328 |
| 16 | SU6 | 12537 | 66 | 4.8066 | 6.4087631088 |

*Figure 57 : Extrait de la table de résultat de l'USURE sur l'échéance T2 2017 pour les crédits à la consommation*

Ceci correspond aux taux publiés du 2ème trimestre 2017 (cf. Tableau 2).

| Catégories | Taux effectif pratiqué **au deuxième trimestre 2017** par les établissements de crédit et les sociétés de financement | Seuil de l'usure applicable à compter du **1er juillet 2017** |
|---|---|---|
| **Contrat de crédit consentis à des consommateurs n'entrant pas dans le champ d'application du 1° de l'article L. 313-1 du code de la consommation ou ne constituant pas une opération de crédit d'un montant supérieur à 75 000 euros destinée à financer, pour les immeubles à usage d'habitation ou à usage professionnel et d'habitation, les dépenses relatives à leur réparation, leur amélioration ou leur entretien.** | | |
| - prêts d'un montant inférieur ou égal à 3000 euros (1): | 15,45% | 20,60% |
| - prêts d'un montant supérieur à 3000 euros et inférieur ou égal à 6000 euros (1): : | 9,84% | 13,12% |
| - prêts d'un montant supérieur à 6000 euros (1): : | 4,80% | 6,40% |

*Tableau 2 : Extrait officiel des taux d'USURE applicables au 1er juillet 2017, http://www.tresor.economie.gouv.fr/7234_seuils-de-l-usure-applicables*

Une fois les données calculées en SAS, il faut les charger dans la base de données ROSTAM.

### 6.5.8 Envoi des résultats à ROSTAM

Une fois les taux d'usure validés par le métier, les données sont transmises à ROSTAM via EAI. L'EAI mis en place prend un fichier ZIP en entrée. J'ai choisi de créer ce fichier avec une procédure stockée SAS. Le contenu de ce fichier sera ensuite chargé dans ROSTAM dans la table V_FCT_FAIT_DIFF.

#### *6.5.8.1 La procédure stockée SAS d'envoi des données*

À l'aide d'une procédure stockée SAS (cf. Figure 58), l'utilisateur crée un fichier ZIP qui sera transféré via EAI vers la base de données ROSTAM. L'utilisateur renseigne uniquement le processus de calcul sur lequel se baser. En effet, plusieurs étapes de calcul peuvent être lancées. Il faut donc en choisir une. Les autres paramètres sont affichés pour information mais ne sont pas modifiables.

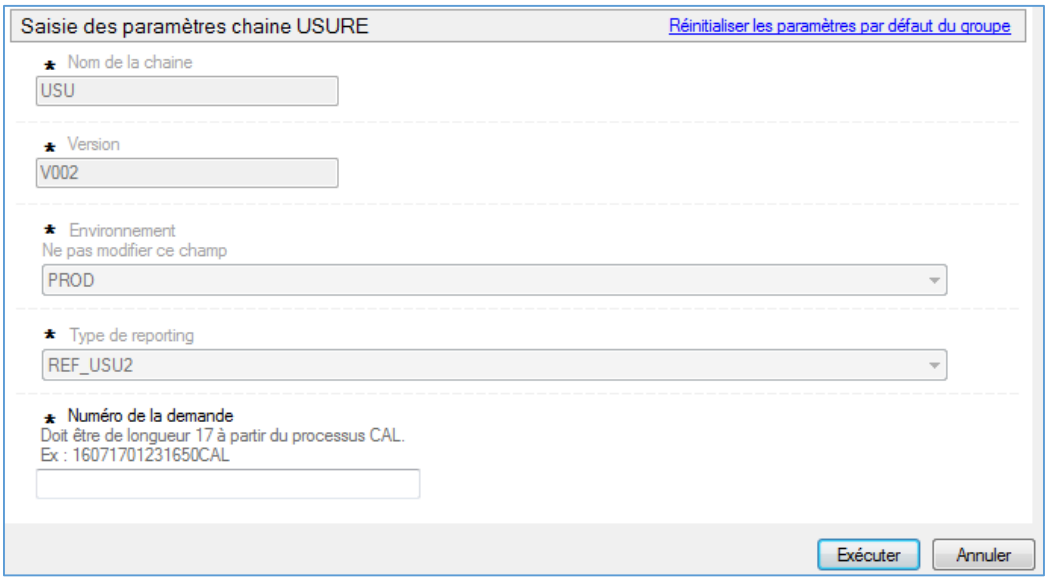


*Figure 58 : Procédure stockée d'envoi des données résultats de l'usure à ROSTAM*

Les paramètres à renseigner lors de l'exécution de l'application stockée sont :

- **Nom de la chaîne** : Le paramètre du nom de la chaîne est figé et donc non modifiable. En effet plusieurs chaînes de traitement utilisent la même procédure stockée ;
- **Version de la chaîne** : Le paramètre de la version de la chaîne USURE est figé et donc non modifiable ;

- **Environnement de la chaîne** : Le paramètre environnement de la chaîne est figé et donc non modifiable. Seule la version de production est active pour l'envoi de données dans ROSTAM. Ce paramètre a été ajouté pour être identique aux autres applications stockées ;
- **Type de Reporting à exporter** : Au 26/01/2017, seuls les résultats des taux d'usure du processus de calcul sont envoyés dans ROSTAM. Le paramètre type de reporting de la chaîne USURE est donc figé et non modifiable. Ce paramètre fait référence à la granularité (enquête) de la chaîne et vaut « USU ». Dans le futur, d'autres données seront envoyées. C'est pourquoi ce paramètre a été créé ;
- **Le numéro de la demande à envoyer** : L'utilisateur doit renseigner le numéro de la demande correspondant au reporting sélectionné préalablement. En effet l'utilisateur peut lancer plusieurs fois la chaîne de calcul mais il doit en choisir une à diffuser. Le numéro de demande est composé de la façon suivante :
    - Échéance (date d'arrêté) sur 4 digits : année, mois. Exemple : 1610 ;
    - Date d'exécution de la demande sur 6 digits : année, mois, jour. Exemple : 170123 ;
    - Heure d'exécution de la demande (chaîne) sur 4 digits : heure, minutes Exemple : 1650 ;
    - Nom du processus émetteur sur 3 caractères : doit être CAL.

Cet exemple correspond à un lancement de chaîne sur l'échéance d'Octobre 2016, exécuté le 23 Janvier 2017 à 16h50.

#### *6.5.8.2 Explication code de l'application stockée*

L'application stockée génère le fichier ZIP à transmettre à ROSTAM et un compte-rendu d'exécution. Je programme alors les tests d'erreur potentielle et affiche des messages clairs à l'utilisateur pour, au besoin, corriger son lancement.

À chaque exécution, j'initialise une table SAS de suivi d'exécution (cf. Figure 59). Cette table est mise à jour si une erreur est détectée.

```
 /* initialisation de la table de suivi des erreurs */
data work.err_msg ;
 length msg $1000 ;
 msg = "----------------------------------------------------------------" ;output ;
 msg = "Début du traitement : %sysfunc(datetime(),is8601dt.)" ;output ;
 msg = "----------------------------------------------------------------" ;output ;
 msg = "" ;output ;
 run ;
```

*Figure 59 : Initialisation de la table SAS de suivi de traitement*

Je crée une macro SAS nommée « CONTROLE » qui contiendra l'ensemble des contrôles.

Le premier contrôle vérifie l'existence du répertoire de la demande. En effet, la saisie de la demande étant manuelle une erreur est facile à commettre. Dans ce cas, le compte-rendu l'identifie (cf. Figure 60) et l'utilisateur vérifie son numéro de demande. L'utilisateur peut consulter la log du traitement dans le chemin mentionné du rapport de la procédure stockée.

```
                                RAPPORT : EXPORT ROSTAM BASE DE SERIES

----------------------------------------------------------------
Début du traitement : 2017-10-14T13:33:55
----------------------------------------------------------------


-->/home/destr/appli/travail/groupes_de_travail/gw_usure_sas/libre/usu_prod/V002/dem/SIRnon_existence_dem/perm n'existe pas

----------------------------------------------------------------
Fin du traitement : 2017-10-14T13:33:55
----------------------------------------------------------------
```

*Figure 60 : Compte rendu de l'exécution de l'application stockée lorsque le répertoire d'entrée n'existe pas*

Je teste l'existence du répertoire en y affectant une librairie SAS (cf. Figure 61). Si sa création est en échec, alors le répertoire n'existe pas. Dans ce cas, j'ajoute dans la table de suivi des messages et j'affecte à la macro variable STATUT la valeur KO1. Cette macro-variable me permet de connaître le résultat des tests.

```
%macro controle;
/* initialisation a vide de la macro variable de controle d'erreur */
%let statut = ;
/* repertoire de stockage des demandes */
%let path = /home/destr/appli/travail/groupes_de_travail/gw_usure_sas/libre/usu_prod/V002/dem/;
/* test de l'existence de la demande */
%if %sysfunc(libname(PERM, &path.SIR&IDDEM./perm)) ne 0 %then %do ;
    %put ** %str(La demande &path.SIR&IDDEM./perm n%'existe pas) **;
    %put ;
    %let statut = KO1;

    proc sql ;
        insert into work.err_msg
        set msg = ""
        set msg = "-->&path.SIR&IDDEM./perm n'existe pas"
        set msg = "";
    quit ;
%end ;
```

*Figure 61 : Code SAS de contrôle d'existence du répertoire de demande*

Dans le cas où le contrôle est passé avec succès, je teste l'existence des tables de résultat du calcul des taux d'usure. Ce cas-là ne peut pas arriver car la chaîne de l'usure est entièrement programmée sans action manuelle. Néanmoins, j'ai opté de le gérer (cf. Figure 62). Si néanmoins cette erreur survient, l'utilisateur vérifiera la log du traitement du calcul des taux d'usure. Il y aura sans doute une erreur empêchant la création de la table résultat.

```
                                RAPPORT : EXPORT ROSTAM BASE DE SERIES

----------------------------------------------------------------------
Début du traitement : 2017-10-14T12:48:06
----------------------------------------------------------------------


-->La table DIFF_ROS_COM n'existe pas dans /home/destr/appli/travail/groupes_de_travail/gw_usure_sas/libre/usu_prod/V002/dem/SIR1701
1708122222CAL/perm


-->La table REF_USU2_BASE_DE_SERIES n'existe pas dans /home/destr/appli/travail/groupes_de_travail/gw_usure_sas/libre/usu_prod/V002/
dem/SIR17011708122222CAL/perm

----------------------------------------------------------------------
Fin du traitement : 2017-10-14T12:52:00
----------------------------------------------------------------------
```

*Figure 62 : Compte rendu de l'exécution de l'application stockée lorsque la table résultat n'existe pas*

Pour vérifier l'existence de la table de résultat, j'utilise la librairie créée lors de mon premier test et la fonction EXIST de SAS qui identifie si un fichier existe (cf. Figure 63). Je réalise alors ce test pour les deux tables résultats de l'usure. Comme pour le premier test, j'ajoute, le cas échéant, l'anomalie dans la table de suivi.

```
/* la demande existence, on verifie l existence des tables résultats */
%else %if "&statut." ne "KO1" %then %do;
    /* test existence de DIFF_ROS_COM */
    %if %sysfunc(exist(PERM.DIFF_ROS_COM)) eq 0 %then %do ;
        %put ** %str(La table DIFF_ROS_COM n%'existe pas dans &path.SIR&IDDEM./perm) **;
        %put ;
        %let statut = KO2;

        proc sql ;
            insert into work.err_msg
            set msg = ""
            set msg = "-->La table DIFF_ROS_COM n'existe pas dans &path.SIR&IDDEM./perm"
            set msg = "";
        quit ;
    %end ;
    /* test existence de REF_USU2_BASE_DE_SERIES */
    %if %sysfunc(exist(PERM.REF_USU2_BASE_DE_SERIES)) eq 0 %then %do ;
        %put ** %str(La table REF_USU2_BASE_DE_SERIES n%'existe pas dans &path.SIR&IDDEM./perm) **;
        %put ;
        %let statut = KO2;

        proc sql ;
            insert into work.err_msg
            set msg = ""
            set msg = "-->La table REF_USU2_BASE_DE_SERIES n'existe pas dans &path.SIR&IDDEM./perm"
            set msg = "";
        quit;
    %end ;
%end;
```

*Figure 63 : Code SAS de contrôle d'existence des tables résultats de l'usure*

Si ces tests sont passés avec succès alors il n'y a pas d'erreur, et le message suivant est affiché dans la log (cf. Figure 64).

```
                                        RAPPORT : EXPORT ROSTAM BASE DE SERIES

-----------------------------------------------------------------------
Début du traitement : 2017-10-14T11:22:43
-----------------------------------------------------------------------


-----------------------------------------------------------------------
TRAITEMENT SANS ERREUR : Export Rostam base de séries terminé
-----------------------------------------------------------------------

-----------------------------------------------------------------------
Fin du traitement : 2017-10-14T11:24:26
-----------------------------------------------------------------------
```

*Figure 64 : Compte rendu de l'exécution de l'application stockée sans erreur*

Je vérifie qu'il n'existe pas d'erreur avec la macro-variable STATUT, qui ne doit pas être égale à KO2 (cf. Figure 65). L'utilisation successive de « ELSE IF » permet de s'assurer que l'ensemble des conditions est respecté et donc qu'aucune anomalie n'a été détectée. Dans ce cas, je crée :

- Les deux fichiers CSV résultats dans la work SAS ;
- Un fichier zip en utilisant la possibilité de SAS à exécuter des commandes systèmes ;
    - X permet d'indiquer à SAS que l'on va passer une ou plusieurs commandes système ;
    - Je me place alors sur ma work avec un CD (Change Directory) ;
    - Je crée un fichier zip contenant les deux fichiers CSV avec la commande ZIP.
- Une ligne dans la table de suivi indiquant qu'il n'y a pas eu d'erreur.

```
%else %if "&statut." ne "KO2" %then %do;
    /* creation des fichiers résultats dans la work */
    data _null_ ;
        set PERM.DIFF_ROS_COM;
        file "%sysfunc(pathname(work))/DIFF_ROS_COM.csv" dlm=";" dsd  termstr=crlf;
        if _n_=1 then put "IDDEM;IDFAIT;IDSOURCE;CODETAP" ;
        put &LIST1. ;
    run ;
    data _null_ ;
        set PERM.REF_USU2_BASE_DE_SERIES ;
        file "%sysfunc(pathname(work))/REF_USU2_BASE_DE_SERIES.csv" dlm=";" dsd  termstr=crlf;
        if _n_=1 then put "IDDEM;ECHEANCE;CODE_CATEG;TRANCHE;TYPE;VALEUR" ;
        put &LIST2. ;
    run ;
    /* creation du zip dans la libraire d'entrée de EAI */
    x "cd %sysfunc(pathname(work)) ;
       zip &LIB_OUT./REF_USU2_BASE_DE_SERIES.zip &DIFF_ROS_COM.csv &REF_USU2_BASE_DE_SERIES.csv" ;
    /* rapport de traitement */
    proc sql ;
        insert into work.err_msg
        set msg = ""
        set msg = "-----------------------------------------------------------------------"
        set msg = "TRAITEMENT SANS ERREUR : Export Rostam base de séries terminé"
        set msg = "-----------------------------------------------------------------------"
        set msg = ""
        set msg = "-----------------------------------------------------------------------"
        set msg = "Fin du traitement : %sysfunc(datetime(),is8601dt.)"
        set msg = "-----------------------------------------------------------------------"
    ;quit ;
%end;
```

*Figure 65 : Code SAS de création des fichiers résultats de l'usure*

Dans un souci de traçabilité des données résultat, les faits utilisés dans le calcul du taux d'usure sont également stockés dans ROSTAM (fichier DIFF_ROS_COM). A mon avis, ceci est inutile, car les faits utilisés sont conservés dans la table des faits. Il y a donc redondance d'information. Néanmoins les deux fichiers sont envoyés à ROSTAM. Ils sont compressés dans un même fichier zip avant l'envoi.

Pour afficher ces rapports de traitement dans la log, j'utilise la fonction « FILE PRINT » de SAS qui affiche le contenu de ma table de suivi (cf. Figure 66).

```
proc sql ;
        insert into work.err_msg
        set msg = "-------------------------------------------------------------------------"
        set msg = "Fin du traitement : %sysfunc(datetime(),is8601dt.)"
        set msg = "-------------------------------------------------------------------------"
        set msg = ""
    ;quit ;

/* écriture dans la log des messages de contrôle */
ods escapechar="^" ;
data _null_ ;
    set work.err_msg end=fin;
    file print ;
    if _n_ = 1 then do ;
        put "^{style[font_size=10pt font_weight=bold font_style=italic ]
             RAPPORT : EXPORT ROSTAM BASE DE SERIES }" ;
        put "^" ;
    end ;
    put msg ;
run ;
```

*Figure 66 : Code SAS d'affiche de la table de suivi du traitement dans la log*

## 6.6 VALIDATION DES RESULTATS

La recette, à l'aide d'une succession de tests, doit valider la conformité par rapport au cahier des charges.

Dans ce projet, il y a deux parties distinctes à valider. La première est celle du passage de la collecte des données dans ROSTAM et de son nouveau système d'information. Je ne participe pas à cette recette. Je ne la détaille donc pas. La seconde partie concerne le fonctionnement de la chaîne SAS de l'usure sur le nouveau système d'information. J'en suis pleinement responsable, avec une validation finale par le métier en charge du calcul du taux d'usure. Aucune règle fonctionnelle n'est modifiée pendant la migration, les résultats de l'ancien système et du nouveau doivent être identiques. Pour valider les résultats il faut alors comparer les résultats des deux méthodes.

Ma méthode de validation des résultats comprend quatre parties :

- Vérification de façon unitaire que les parties modifiées sont correctes ;
- Vérification que les données extraites sont identiques selon les deux méthodes ;
- Vérification que les données de calcul sont identiques selon les deux méthodes ;
- Vérification que le stockage des données résultats dans ROSTAM est correct.

Je crée un cahier de tests (cf. Tableau 3) afin de tracer et structurer ma recette. Il contient les tests que je juge pertinents à réaliser. J'y associe un programme SAS permettant d'exécuter chacun de ces tests.

| Numéro test | Test | Programme | Remarque | Résultat |
|---|---|---|---|---|
| 1 | Les extractions de ROSTAM et de SISMF doivent être identiques. | SIR_CHA.SAS | Sur 1,3 millions de lignes, environ cinq cent ont une différence. L'impact est faible et le métier accepte ce delta. | **OK** |
| 2 | L'identifcation des caractéristiques des catégories de l'usure dépend de la date d'échéance | SIR_CHA.SAS | | **OK** |
| 7 | Par défaut, la chaîne est lancée complétement. | STP_SIR_MAIN.SAS | | **OK** |
| 18 | Les données résultats stockées dans ROSTAM sont conformes aux données résultats. Comparer les données de la SIR_CAL.REF_USU2 et celles de ROSTAM. | N/A | | **OK** |
| 22 | Le temps d'éxcution de la chaîne compléte est inférieure à 30 minutes | N/A | | **OK** |

*Tableau 3 : Extrait du cahier de test de la chaîne de calcul des taux d'usure*

En environnement de recette je réalise les tests unitaires vérifiant que mon code fonctionne correctement. Je n'y ai pas réalisé de comparaison entre les deux systèmes car les données entre les deux systèmes sont différentes. En effet, la MOE en charge des tests de passage de la collecte dans ROSTAM n'ont pu charger que des sous-ensembles de données pour cause de manque de volumétrie sur cet environnement.

En environnement d'intégration, l'ensemble des données sur les échéances ciblées sont bien présentes. Je peux réaliser l'intégralité de mon cahier de tests. J'exécute les lancements sur les périodes cibles (juillet 2016, octobre 2016). Il y aura également un « double run » en production ancien et nouveau système en production sur l'échéance d'avril 2017 afin de s'assurer que la livraison en production est conforme à l'attendu.

J'ai détecté des erreurs de chargement des données de collecte de l'usure. Je ne devais pas à avoir à valider cette partie, mais vu que mes résultats étaient dépendants des données en

entrée, j'avais positionné un test de comparaison des données. Des règles différentes de contrôle d'intégration des fichiers avaient été implémentées dans ROSTAM par rapport à l'existant. Un grand nombre de données étaient alors rejetées à tort et cela a eu logiquement pour effet d'obtenir un important delta dans le chargement. Une fois l'anomalie corrigée, j'ai pu comparer les résultats.

La procédure « PROC COMPARE » compare une table SAS originale (base) avec une deuxième table SAS (compare). Ayant conservé le même formalisme des tables SAS dans la chaîne de l'usure, cette procédure me permet de comparer aisément les résultats des deux systèmes. Le code ci-dessous (cf. Figure 67) compare la table résultat des taux d'usure (REF_USU2) de l'échéance d'avril 2017 sur la clé du code de catégorie du taux d'usure (CODE_CATEG).

```
proc compare base    = sir_calo._17041707181623_REF_USU2
             compare = sir_cal._17041707181421_REF_USU2
             out     = difference
             outbase outcomp outdiff outnoequal;
    id code_categ;
run;
```

*Figure 67 : Chaîne SAS de l'usure - code SAS comparaison résultat*

Le résultat de cette procédure est ci-dessous (cf. Figure 68). Les deux tables du haut de la figure sont les tables comparées et en dessous le résultat de la procédure PROC COMPARE.

**Avant modification (SISMF, type = base)**

| | code_categ | MOY_MONT | MOY_DUREE | TX_USU |
|---|---|---|---|---|
| 1 | 3A6 | 4410 | 45 | 13.123891904 |
| 2 | AI21 | 920275 | 14 | 1.1090024871 |
| 3 | AI22 | 26010 | 7 | 2.2125824809 |
| 4 | DEC1 | 354 | . | 15.409252413 |
| 5 | DEC2 | 4926 | . | 13.689498099 |
| 6 | IA3 | 346 | 13 | 20.603103328 |
| 7 | IMF1 | 77721 | 85 | 3.1481193721 |
| 8 | IMF2 | 118507 | 171 | 3.1241331147 |
| 9 | IMF3 | 155227 | 272 | 3.2519108953 |
| 10 | IMR | 174825 | 22 | 3.3497324398 |
| 11 | IMV | 136559 | 194 | 2.9358024847 |
| 12 | S2F1 | 416117 | 99 | 1.2317273522 |
| 13 | S2F2 | 96766 | 95 | 2.4538442919 |
| 14 | S2V1 | 3166844 | 173 | 0.7209033544 |
| 15 | S2V2 | 160825 | 90 | 2.2201555 |
| 16 | SU6 | 12533 | 66 | 6.4082261924 |
| 17 | VAT1 | . | 85 | |
| 18 | VAT2 | 3066 | 5 | 4.1635516992 |

**Après modification (ROSTAM, type = compare)**

| | code_categ | moy_mont | moy_duree | tx_usu |
|---|---|---|---|---|
| 1 | 3A6 | 4410 | 45 | 13.123891904 |
| 2 | AI21 | 920275 | 14 | 1.1090024871 |
| 3 | AI22 | 26010 | 7 | 2.2125824809 |
| 4 | DEC1 | 354 | . | 15.409252413 |
| 5 | DEC2 | 4926 | . | 13.689498099 |
| 6 | IA3 | 346 | 13 | 20.603103328 |
| 7 | IMF1 | 77721 | 85 | 3.1481193721 |
| 8 | IMF2 | 118510 | 171 | 3.1241197818 |
| 9 | IMF3 | 155228 | 272 | 3.2519028187 |
| 10 | IMR | 174825 | 22 | 3.3497324398 |
| 11 | IMV | 136559 | 194 | 2.9358024847 |
| 12 | S2F1 | 416117 | 99 | 1.2317273522 |
| 13 | S2F2 | 96756 | 95 | 2.4539700724 |
| 14 | S2V1 | 3166844 | 173 | 0.7209033544 |
| 15 | S2V2 | 160825 | 90 | 2.2201555 |
| 16 | SU6 | 12537 | 66 | 6.4087631088 |
| 17 | VAT1 | . | 85 | . |
| 18 | VAT2 | 3047 | 6 | 4.1631041149 |

**Comparaison des résultats**

| | _TYPE_ | code_categ | MOY_MONT | MOY_DUREE | TX_USU |
|---|---|---|---|---|---|
| 1 | BASE | IMF2 | 118507 | 171 | 3.1241331147 |
| 2 | COMPARE | IMF2 | 118510 | 171 | 3.1241197818 |
| 3 | DIF | IMF2 | 3 | 0 | -0.000013333 |
| 4 | BASE | IMF3 | 155227 | 272 | 3.2519108953 |
| 5 | COMPARE | IMF3 | 155228 | 272 | 3.2519028187 |
| 6 | DIF | IMF3 | 1 | -0 | -8.076616E-6 |
| 7 | BASE | S2F2 | 96766 | 95 | 2.4538442919 |
| 8 | COMPARE | S2F2 | 96756 | 95 | 2.4539700724 |
| 9 | DIF | S2F2 | -9 | -0 | 0.0001257805 |
| 10 | BASE | SU6 | 12533 | 66 | 6.4082261924 |
| 11 | COMPARE | SU6 | 12537 | 66 | 6.4087631088 |
| 12 | DIF | SU6 | 4 | -0 | 0.0005369164 |
| 13 | BASE | VAT2 | 3066 | 5 | 4.1635516992 |
| 14 | COMPARE | VAT2 | 3047 | 6 | 4.1631041149 |
| 15 | DIF | VAT2 | -19 | 2 | -0.000447584 |

*Figure 68 : Chaine SAS de l'usure - comparaison des résultats de l'échéance avril 2017*

Il y a donc cinq catégories (sur les dix-huit) pour lesquelles il y a une différence de résultat. Ceci est dû aux quelques différences des données chargées entre l'ancien et le nouveau système. Ce sont des différences mineures acceptées par le métier.

Remarque : le libellé des catégories de crédit des taux d'usure est en annexe (cf. Tableau 7).

Une fois mon cahier de test entièrement validé, j'ai donné la main au métier pour la vérification d'aptitude au bon fonctionnement (VABF). Cette dernière s'est correctement déroulée. Elle m'a permis d'améliorer le manuel utilisateur d'utilisation de la chaîne de l'usure afin qu'il soit le plus clair possible.

Le respect du procédé établi pour la recette a permis de mettre en production la partie SAS sans anomalie.

## 6.7 CHARGE POUR LES EVOLUTIONS SAS

À l'issue de cette analyse, j'avais estimé le coût des évolutions SAS à 105 jours :

- Analyse 5j ;
- Conception- Spécifications techniques 15j ;
- Implémentation 70j ;
- Recette 15j ;

Les estimations de charge d'analyse, de conception et d'implémentation ont été correctes à quelques jours près.

Par contre la phase de recette a été plus longue que les 15 jours estimés. En effet, le delta de chargement des données dans ROSTAM a augmenté le coût de la recette. J'ai vérifié beaucoup plus de choses que prévu. Le temps de recette a été de 30 jours.

Il y a eu en plus 10j pour la coordination de l'ensemble de la migration et pour la réflexion sur le nouveau modèle de données.

Le total a donc été de 130 jours/homme pour les évolutions SAS.

## 6.8 DISCUSSION DES CHOIX

La migration partielle de la chaîne décisionnelle de l'usure a correctement fonctionné. Mais selon moi, des choix erronés ont été faits lors de la conception du projet. Le choix d'utilisation de l'EAI, la mise en œuvre du modèle en étoile et des définitions de variables non adaptées dans ROSTAM sont les principaux points que je remets en cause.

### 6.8.1 Le choix de l'EAI pour le transfert des données de SAS à ROSTAM

Selon moi, le choix de l'utilisation de EAI n'est pas justifié. Il a été un centre de coût pour l'intégration des données. Les deux principales raisons de ce choix pour l'OI étaient que :

- Les utilisateurs ne doivent pas pouvoir écrire dans la base de données ;
- Les données doivent être contrôlées avant insertion en base de données.

Sans m'opposer à ces raisons, j'ai conseillé de créer une table « temporaire » dans la base de données ROSTAM où les utilisateurs SAS de la chaîne de l'usure auraient accès en écriture. En effet, il est possible de créer des droits différents sur les tables d'une base de données

avec des « grants ». Une fois les données insérées dans cette table, il est possible de les contrôler avant de les insérer dans la table finale (avec un trigger ou autre).

La seule contrainte supplémentaire est que les applications SAS et ROSTAM doivent être simultanément disponibles. C'est de toute façon nécessaire car SAS utilise très souvent la base de données ROSTAM. On peut donc considérer cette contrainte comme inexistante. On s'est alors privé de l'utilisation du chargement en base de données que permet un ETL.

### 6.8.2 La mise en œuvre du modèle étoile de la base de données ROSTAM

Un des objectifs du projet ROSTAM est d'obtenir un unique système d'information stockant les informations de collectes et de résultats des chaînes de production statistiques de la DGS. Le choix de conception a été d'utiliser un modèle en étoile. Selon moi, plusieurs erreurs de conceptions ont été faites dont je vais détailler les principales.

#### *6.8.2.1 Principe de conception du modèle en étoile non respecté*

La mise en œuvre du modèle en étoile ne respecte pas les principes de ce modèle (cf. chapitre 5.2.9.3 ROLAP).

La table de fait contient, en plus des ID des dimensions, les informations de tables de ces mêmes tables de dimension (cf. Figure 69). C'est le cas de l'échéance de la donnée. La table de fait stocke l'identifiant de l'échéance (ECH_ID) pour réaliser la jointure avec la table dimension. Ceci est conforme à la théorie du modèle en étoile, mais elle contient également la valeur de cette échéance (ECHEANCE_FAIT). Dans certaines tables stockant une échéance il n'y a pas la variable identifiant de l'échéance (table V_FCT_FAIT_DIFF). Il n'y a donc pas d'homogénéité dans les tables.

Il y a également des informations explicatives stockées directement dans la table de fait sans avoir de table de dimension associée. C'est par exemple le cas de l'information du CIB. Cette information devrait être stockée dans une table de dimension et la table de fait stocker uniquement l'identifiant. Je parle plus loin dans ce chapitre des conséquences de ce choix sur les performances.

| Nom | Type | Longueur | Format | Informat | Libellé |
|---|---|---|---|---|---|
| AJUST | Alphanumérique | 1 | $1. | $1. | AJUST |
| ANOMALIE_APRES_FILTRE | Alphanumérique | 30 | $30. | $30. | ANOMALIE_APRES_FILTRE |
| ANOMALIE_CONTROLE_SPECIFIQUE | Alphanumérique | 30 | $30. | $30. | ANOMALIE_CONTROLE_SPECIFIQUE |
| ANOMALIE_FAIT_DUPLIQUE | Alphanumérique | 30 | $30. | $30. | ANOMALIE_FAIT_DUPLIQUE |
| ANOMALIE_FORMULE | Alphanumérique | 30 | $30. | $30. | ANOMALIE_FORMULE |
| ANOMALIE_SIREN | Alphanumérique | 30 | $30. | $30. | ANOMALIE_SIREN |
| CAP | Numérique | 8 | 7. | 7. | CAP |
| CDT_NGCT | Alphanumérique | 1 | $1. | $1. | CDT_NGCT |
| CIB | Alphanumérique | 255 | $255. | $255. | CIB |
| CODE_DEVISE | Alphanumérique | 10 | $10. | $10. | CODE_DEVISE |
| CODE_GUICHET | Alphanumérique | 10 | $10. | $10. | CODE_GUICHET |
| CODE_NATIONAILTE | Alphanumérique | 10 | $10. | $10. | CODE_NATIONAILTE |
| CODE_PAYS | Alphanumérique | 10 | $10. | $10. | CODE_PAYS |
| CODE_SISMF | Alphanumérique | 255 | $255. | $255. | CODE_SISMF |
| CODE_SOURCE | Alphanumérique | 255 | $255. | $255. | CODE_SOURCE |
| CONF_STATUS | Alphanumérique | 2 | $2. | $2. | CONF_STATUS |
| CONTROLE_SPECIFIQUE_APRES_FIL | Alphanumérique | 30 | $30. | $30. | CONTROLE_SPECIFIQUE_APRES_FILTRE |
| COURS_TITRE | Numérique | 8 | | | COURS_TITRE |
| CSI_Id | Numérique | 8 | 20. | 20. | CSI_Id |
| DATE_REMISE_OG | Date | 8 | DATE | DATE | DATE_REMISE_OG |
| DATE_TRT_DEM | Date | 8 | DATE | DATE | DATE_TRT_DEM |
| DEC_Id | Numérique | 8 | 20. | 20. | DEC_Id |
| DUREE_IN | Numérique | 8 | 4. | 4. | DUREE_IN |
| ECH_Id | Numérique | 8 | 20. | 20. | ECH_Id |
| ECHEANCE_FAIT | Date | 8 | DATE | DATE | ECHEANCE_FAIT |
| ETA_Id | Numérique | 8 | 20. | 20. | ETA_Id |
| FAIT_NON_NUMERIQUE | Alphanumérique | 500 | $500. | $500. | FAIT_NON_NUMERIQUE |
| FCT_Id | Numérique | 8 | 20. | 20. | FCT_Id |
| FCT_Idligne | Numérique | 8 | 20. | 20. | FCT_Idligne |
| FRQ_Id | Numérique | 8 | 20. | 20. | FRQ_Id |
| IDENTIFIANT_TITRE | Alphanumérique | 12 | $12. | $12. | IDENTIFIANT_TITRE |
| IDX_REF | Numérique | 8 | 2. | 2. | IDX_REF |
| INS_FI | Alphanumérique | 3 | $3. | $3. | INS_FI |
| LAST_VERSION | Numérique | 8 | 1. | 1. | LAST_VERSION |
| MONNAIE | Alphanumérique | 3 | $3. | $3. | MONNAIE |
| MONTANT | Numérique | 8 | | | MONTANT |

*Figure 69 : Extrait du format de la table V_FCT_FAIT*

#### *6.8.2.2 La table de dimension TEMPS*

Le choix de conception de table de dimension TEMPS, soit dans notre modèle la table V_DIM_ECHEANCE, est contraignante. Elle ne stocke pas de date mais uniquement deux champs numériques MOIS et ANNEE. Outre le fait de ne pas pouvoir effectuer un filtre avec une date, cela impose que l'ensemble des collectes soit de périodicité mensuelle ou du moins de mois « plein » (trimestrielle, semestrielle, etc.). Si une collecte est bimensuelle,

alors le formalisme de la table n'est plus adapté. J'aurai créé une dimension TEMPS en stockant un « calendrier » dont chaque jour aurait un identifiant unique. Cela n'aurait pas eu d'impact sur les performances car les variables numériques de faible valeur prennent peu de place mémoire. La volumétrie augmenterait (une ligne par jour versus une ligne par mois dans la table de dimension TEMPS) mais elle serait quand même d'un faible volume. Cela permettrait d'avoir n'importe quelle périodicité de collecte et d'avoir clairement une date et non une concaténation de variable.

#### *6.8.2.3 Des noms de variable sans sens*

Des variables n'ont pas « de sens ». Par exemple, que contient la variable « FAIT_NON_NUMERIQUE » dans la table de fait (cf. Figure 69) ? On comprend que cela sera du texte mais il est quand même préférable de créer des variables stockant une information précise et non un type de format.

#### *6.8.2.4 Une table de fait unique*

Il y a également, selon moi, une erreur sur le choix de créer une unique table de fait. La grande majorité des collectes de données servant aux chaînes de production sont stockées dans la table de fait unique V_FCT_FAIT. Pour autant :

- La majorité des indicateurs de chacune des collectes est différent. Il a alors été ajouté des indicateurs particuliers pour chacune des collectes. ;
- La périodicité est différente selon les collectes. Il y a des collectes mensuelles, trimestrielles…Une des caractéristiques des tables de fait est que le fait doit être défini sur une périodicité unique (le nombre de visites d'un musée par jour, le nombre de réparations de voiture par mois, etc.). Dans notre cas, il n'y a donc pas de périodicité précise.

Ce ne sont que des exemples parmi d'autres. Outre ce problème de conception, il y a également un problème de définition des formats des champs.

### 6.8.3 Des définitions de variables non adaptées dans la base de données ROSTAM

Dans la base de données ROSTAM, beaucoup de variables sont définies avec un format de stockage plus grand que nécessaire. Un stockage plus grand entraîne une table plus volumineuse et donc une table plus lente à interroger. Je commencerai par un bref rappel sur le calcul de la taille d'une table puis je montrerai quelques exemples de formats non adaptés dans ROSTAM.

#### *6.8.3.1 Rappel sur la taille d'une table*

La taille d'une table est principalement liée à la taille et au nombre d'occurrences de ses tuples. La taille d'une table peut être comptabilisée en nombre de « page ou bloc ». La taille d'un bloc est généralement entre 1Ko et 8Ko. Plus les tuples prennent de l'espace mémoire, moins il y en aura sur un bloc et donc plus il y aura de blocs. La taille d'un tuple se calcule de la façon suivante :

- 1 caractère (ascii) = 1 octet ;
- Un entier de -128 (-2^7) à 127 (2^7 - 1) = 1 octet ;
- Un entier de -32 768 (-2^15) à 32 767 (2^15 -1) = 2 octets ;
- Un entier non signé de 65 537 à 16 777 216 (2^24) = 3 octets ;
- …

Remarque : Je prends l'exemple des entiers car je parlerai de longueur trop importante d'ID de tables de dimension. Ces ID sont des entiers non signés, mais malgré mes recherches je n'ai pas trouvé de format « non signé » en SQLSERVER (type de la base de données ROSTAM).

L'exemple ci-dessous (cf. Figure 70), montre la différence en nombre de pages pour deux tables SAS de 100 000 observations. Les deux tables contiennent exactement les mêmes tuples mais dans la première (test10) les variables sont définies sur une longueur de dix caractères et dans la deuxième (test100) sur une longueur de cent.

```
23          DATA test10;
24           LENGTH var var2 $10;
25           var = 'texte1'; var2 = 'texte2';
26           DO i = 1 TO 100000;OUTPUT;END;
27          RUN;

NOTE: The data set WORK.TEST10 has 100000 observations and 3 variables.
NOTE: Compressing data set WORK.TEST10 increased size by 40.00 percent.
      Compressed is 70 pages; un-compressed would require 50 pages.
NOTE: DATA statement used (Total process time):
      real time           0.07 seconds
      cpu time            0.08 seconds


28          DATA test100;
29           LENGTH var var2 $100;
30           var = 'texte1'; var2 = 'texte2';
31           DO i = 1 TO 100000;OUTPUT;END;
32          RUN;

NOTE: The data set WORK.TEST100 has 100000 observations and 3 variables.
NOTE: Compressing data set WORK.TEST100 decreased size by 77.12 percent.
      Compressed is 73 pages; un-compressed would require 319 pages.
NOTE: DATA statement used (Total process time):
      real time           0.09 seconds
      cpu time            0.10 seconds
```

*Figure 70 : Exemple d'impact de la taille de variable sur le nombre de bloc d'une table*

Sur un format de table non compressée, la première table est constituée de 50 blocs et la seconde de 319. La seconde table est donc six fois plus volumineuse alors qu'elle contient exactement les mêmes données.

L'impact est donc important sur la taille mais également sur la performance d'interrogation. En effet, sans index, l'interrogation d'une table se fait en parcourant l'intégralité des blocs. L'interrogation de la seconde table, sans index, sera donc plus lente.

Beaucoup de variables caractères sont définies avec des longueurs plus grandes que nécessaire. Je prendrai uniquement des exemples de la table des faits car c'est celle où l'impact de ces définitions de format non adaptées sera le plus important.

#### *6.8.3.2 Des définitions de variables non adaptées dans ROSTAM*

La variable CIB, outre le fait qu'elle ne devrait être présente dans la table de fait, est strictement sur 9 caractères. Or, elle est définie sur 250 dans la table. Il en est de même pour

la variable CODE_SISMF qui définit le code des séries calculées. Elle est au maximum de 50 caractères mais elle est également définie sur 250.

Il en va de même pour les variables « ID » des dimensions. Elles sont toutes définies avec le format « BIGINT », soit un stockage sur huit octets de type (cf. Tableau 4).

| Type de données | Plage | Stockage |
|---|---|---|
| **bigint** | -2^63 (-9,223,372,036,854,775,808) à 2^63-1 (9,223,372,036,854,775,807) | Huit octets |
| **int** | -2^31 (-2 147 483 648) à 2^31-1 (2 147 483 647) | Quatre octets |
| **smallint** | -2^15 (-32 768) à 2^15-1 (32 767) | Deux octets |
| **tinyint** | 0 à 255 | Un octet |

*Tableau 4 : Format numérique de type entier SQLSERVER, https://docs.microsoft.com/fr-fr/sql/t-sql/data-types/int-bigint-smallint-and-tinyint-transact-sql*

Par exemple, la table de dimension V_DIM_SOURCE stocke les quatre cents types de sources de données différentes pour la DGS. L'identifiant maximum est donc de 400. Un stockage de type « SMALLINT » suffit amplement. Cela diviserait par quatre la taille de stockage de cette variable dans les tables V_DIM_SOURCE (faible impact) et V_FCT_FAIT (fort impact).

### 6.8.4 Les contrôles SAS d'envoi des données à ROSTAM

Lors de la rédaction de ce document, je me suis aperçu que je n'avais pas testé la bonne création des fichiers ZIP à destination de ROSTAM. En effet, je vérifie que les données nécessaires sont présentes pour créer le fichier ZIP mais pas sa création finale. Il manque donc un test d'existence du fichier ZIP. Je l'ai ajouté après la rédaction de ce document.

# 7 PROPOSITION DE METHODOLOGIE DE MIGRATION DE SAS VERS R

Face à un usage croissant des langages de programmation open-source R et Python au sein de la communauté statistique et à une amélioration des solutions de mise à disposition de ces langages, la Banque de France souhaite mettre en place des environnements de travail adaptés à ces langages. L'intégration de ces outils renforcera la palette d'outils à disposition des analystes et développeurs pour la réalisation ou le prototypage d'outils analytiques. En pratique, ces environnements seront également une des solutions à disposition des analystes souhaitant migrer des productions (chaines statistiques, analyses, …) réalisées jusqu'à maintenant dans un environnement SAS. Le logiciel SAS est utilisé depuis de nombreuses années à la DGS. Il y a donc un fort historique sur cette technologie. La DGS s'oriente dans un premier temps vers une plus grande utilisation du logiciel R.

Il y a donc la problématique de reprise de l'historique des traitements SAS dans le langage R. La problématique qui m'a été soumise est « combien coûterait la migration des traitements SAS en R de la DGS ? ». J'ai alors réfléchi à une proposition méthodologique d'estimation de coût de migration des programmes SAS vers R.

## 7.1 CONCEPTION D'UN OUTIL D'ESTIMATION DE CHARGE DE MIGRATION

Pour répondre au besoin d'estimation de la direction de la DGS, j'ai réfléchi à une méthode qui estimerait automatiquement le coût de migration d'un ou plusieurs programmes SAS.

### 7.1.1 Le concept général

Le postulat de départ est que le temps de mise en place d'un traitement est identique en SAS ou en R (ou même dans une autre technologie similaire). Certains traitements seront plus longs ou un plus rapides à développer en R par rapport à SAS mais en moyenne cela s'équilibrera. Avec ce postulat, il est juste de se demander pourquoi essayer de calculer le coût de migration R s'il est identique au coût de mise en œuvre avec SAS. La réponse est simple, le coût de mise en œuvre est dans beaucoup de cas, inconnu ou oublié. Il faut donc le calculer.

La question vaut également pour le périmètre pris en compte dans l'estimation de charge. Est-ce uniquement la charge technique de « codage » ou plus largement l'ensemble des tâches (spécifications, recette, etc.) ? Sur une migration sans évolution fonctionnelle, les

spécifications fonctionnelles (si elles existent) ne seront normalement pas à modifier. Mais, à minima, il sera nécessaire de vérifier leurs cohérences avec la version actuelle car il est possible que ces documents ne soient pas à jour. Les spécifications techniques seront à modifier vu que l'on change de langage. La phase de recette sera, elle aussi, à réaliser. S'il y a une évolution fonctionnelle alors l'ensemble les spécifications fonctionnelles et techniques seront à mettre à jour. J'ai donc pris le postulat que, pour réaliser la migration d'un traitement, le temps à consacrer sera identique à son temps initial de mise en place (spécification fonctionnelle, spécification techniques, développement et recette). Ceci n'est sans doute pas exact à 100% car il y a beaucoup d'autres paramètres jouant sur le temps de migration (niveau d'expertise de la personne, qualité du code, qualité des spécifications…). Mais l'objectif n'est pas de donner un chiffre précis mais bien un ordre de grandeur. Dans cette optique mon postulat me semble correct.

Je conceptualise alors un outil scannant le ou les traitements à migrer pour obtenir une estimation de temps de charge de migration et une identification de complexité. Je détermine plusieurs indicateurs techniques que je juge pertinents. Le schéma ci-dessous (cf. Figure 71) explique l'idée générale. Plus on rajoute des « fonctionnalités », plus la complexité identifiée (et donc son coût de migration) est élevée.

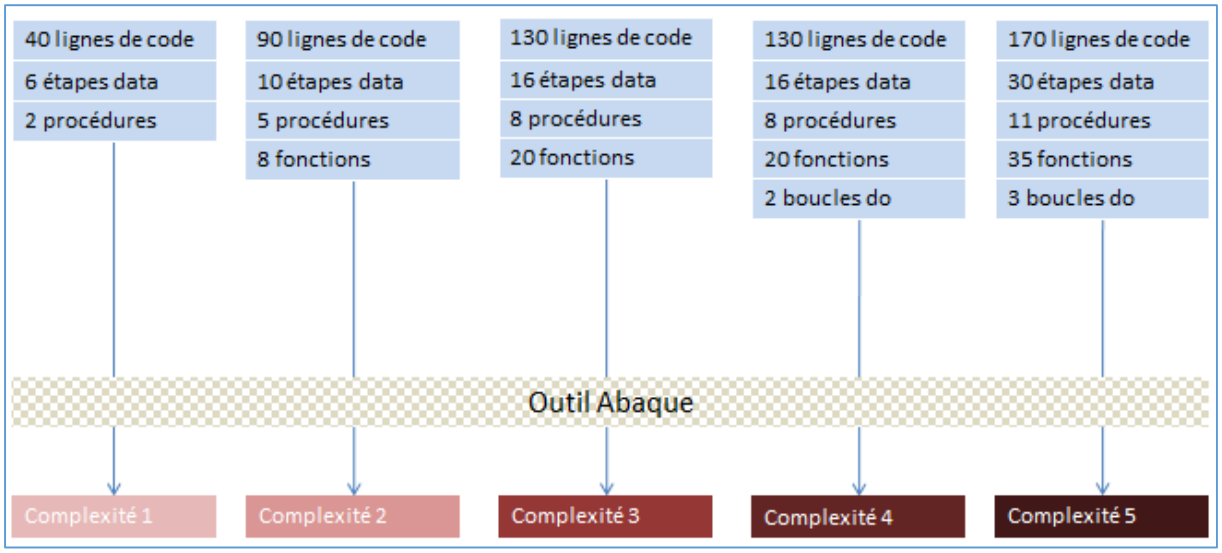


*Figure 71 : Établir la complexité de migration avec un abaque*

Ce schéma montre les premiers indicateurs identifiés. Je vais donc détailler ces choix.

### 7.1.2 Définir les indicateurs pour réaliser l'estimation de charge

En scannant informatiquement le ou les programmes à migrer, j'essaye de déterminer la charge homme de migration et la complexité du traitement. Je prends en compte deux faits qui me paraissent justes :

- Toute chose égale par ailleurs, plus un programme est long, plus il y a de calculs, plus il est complexe à migrer ;
- Plus il y a de fonctions différentes dans le programme plus il est complexe à migrer.

J'ai choisi les indicateurs suivants en réfléchissant sur les différences de programmation de SAS et de R :

- Le nombre d'étape « data » :
    - SAS fonctionne en étape « DATA » qui est en quelque sorte un « mini traitement » sur des données ;
    - Connaître le nombre d'étape DATA est donc, selon moi, un indicateur important pour réaliser l'estimation.
- Le nombre de procédure SAS :
    - SAS propose de nombreuses procédures qui permettent de réaliser des moyennes, des tests statistiques, des graphiques, etc. ;
    - Connaître le nombre de procédure SAS est donc, selon moi, un indicateur important pour réaliser l'estimation.
- L'ensemble des fonctions :
    - Plus il y a de fonctions différentes dans un programme plus il sera complexe à migrer ;
    - Plus une fonction est utilisée dans un programme, plus on apprend à la mettre en œuvre et donc plus on le fait rapidement.
- Le nombre de boucles de programmation :
    - SAS permet de créer automatiquement des lignes de code SAS avec des boucles de macro langage (exemple avec les boucles DO) ;
    - SAS permet également de créer une boucle de traitement sur chaque ligne d'une table ;
    - Cela est, pour moi, une complexité à identifier et à prendre en compte.

- Le nombre de lignes hors commentaire (cet indicateur a ensuite été transformé en nombre d'instruction SAS) :
    - Plus un programme contient des lignes de code, plus il sera long à migrer ;
    - Un programme de dix lignes sera normalement moins long à migrer qu'un programme de mille lignes.

Ces indicateurs servent à calculer la note de complexité et à estimer la charge de migration.

### 7.1.3 Établir la note de complexité et l'estimation de charge

J'associe à chacun des indicateurs une pondération. La multiplication du nombre d'occurrences de l'indicateur et de sa pondération établit une note de complexité. Pour l'indicateur sur les fonctions, j'utilise la fonction racine carré et non une pondération « fixe ». En effet, plus une fonction est utilisée moins elle devient complexe et coûteuse à mettre à en œuvre. La fonction racine carré correspond à ce comportement. Par exemple :

- Si on utilise une fois une fonction, le score est √1 soit 1 ;
- Si on utilise deux fois la même fonction, le score est √2 soit 1,4 ;
- Si on utilise quatre fois la même fonction, le score est √4 soit 2 ;
- Si on utilise seize fois la même fonction, le score est √16 soit 4 ;
- …

Pour les autres indicateurs, l'utilisation d'une pondération fixe me semble appropriée.

En sommant l'ensemble des notes de complexité des indicateurs, j'obtiens un score de complexité. Ce dernier donne une complexité (de 1 à 5) et un intervalle de nombre de jours/homme estimé pour migrer le programme de SAS vers R.

Lors des premiers lancements de l'abaque, période que je nomme « avant apprentissage », quatre indicateurs sont définis (cf. Tableau 5). J'ai établi ces pondérations, intervalles de complexité et d'estimation de charge à l'aide de plusieurs programmes tests dont je connaissais ces paramètres. Après avoir obtenu des résultats satisfaisants, j'ai commencé ma phase « d'apprentissage ».

La phase d'apprentissage est une utilisation d'environ 150 programmes dont je connais, soit la charge globale de mise en œuvre (charge totale du projet et non d'un programme

unitaire), soit la charge unitaire du programme. Cela m'a permis d'identifier des lacunes, notamment sur la prise en compte des fonctions SAS, et j'ai modifié les intervalles afin que les estimations correspondent au plus près de la réalité.

**AVANT APPRENTISSAGE**

| Indicateur | Pondération |
|---|---|
| 50 lignes de code | 2 |
| Etape data | 1 |
| Procédure | 1 |
| Boucle | 3 |

| Score | Complexité | NB Jour |
|---|---|---|
| 1 - 10 | 1 | 1 - 2 |
| 11 - 20 | 2 | 2 - 4 |
| 21 -40 | 3 | 4 - 8 |
| 41 - 60 | 4 | 8 - 12 |
| > 60 | 5 | > 12 |

**APRES APPRENTISSAGE**

| Indicateur | Pondération |
|---|---|
| Une instruction SAS | 0,1 |
| Etape data | 1 |
| Procédure | 1 |
| Boucle | 2 |
| Fonctions SAS | √(x) |

APPRENTISSAGE

| Score | Complexité | NB Jour |
|---|---|---|
| 1 - 22 | 1 | 1 - 2 |
| 23 - 55 | 2 | 2 - 5 |
| 56 - 110 | 3 | 5 - 10 |
| 111 - 165 | 4 | 10 - 15 |
| > 165 | 5 | >15 |

*Tableau 5 : Pondération et score de complexité de l'abaque de migration*

J'établis une correspondance quasi linéaire entre le score de complexité et le nombre de jours/homme estimé. Une unité du score de complexité vaut 0,091 jour/homme de charge estimée. Je n'ai pas établi ce lien de 0,091 par formule scientifique mais uniquement par observation des résultats de mon échantillon d'apprentissage pour qu'ils correspondent à la réalité. Pour un score de complexité de 22, il suffit d'appliquer la multiplication par 0,091 pour trouver 2 jours/homme de charge estimée (environ).

### 7.1.4 Choix technique

Un programme SAS liste le contenu d'un répertoire (paramètre en entrée du traitement) à analyser. Sur chacun de ces fichiers, un traitement Shell est exécuté pour le scanner, calculer les occurrences de chaque indicateur et créer un fichier de résultat. SAS importe ces informations pour établir la complexité finale.

## 7.2 IMPLEMENTATION DE L'OUTIL D'ESTIMATION

Je réalise l'implémentation des parties de l'outil. Je délègue la partie SHELL à une personne de mon équipe. Je détaillerai donc plus succinctement cette partie.

Pour rappel, l'enchaînement technique est SAS –SHELL – SAS et plus précisément :

1- SAS : identification des programmes à analyser ;
2- SHELL : lancement en boucle du SHELL d'analyse des indicateurs sur chaque programme ;
3- SAS : Importation des résultats du SHELL et calcul de l'estimation ;
4- SAS : Mise en forme des résultats.

### 7.2.1 Identification des programmes à analyser

Le traitement SAS a trois paramètres en entrée :

- REP_IN : le répertoire à analyser ;
- REP_OUT : le répertoire de stockage des résultats ;
- TAB_OUT : Le préfixe des tables résultats.

Après avoir vérifié l'existence du répertoire à analyser (à l'identique que pour la procédure stockée d'export des résultats des données de l'usure dans ROSTAM), je liste les fichiers du répertoire de façon récursive (cf. Figure 72). Je fais cette recherche principalement avec les fonctions Shell suivantes (SAS peut exécuter des commandes système) :

- FIND permet de chercher les fichiers dans un répertoire ;
- LS liste l'ensemble des arguments d'un fichier. J'utilise l'option –R pour la récursivité.

```
/home/destr/appli/travail/groupes_de_travail/gw_cft/libre/cft_save/v100/prog/bao/cft_affec_lib.bps
/home/destr/appli/travail/groupes_de_travail/gw_cft/libre/cft_save/v100/prog/bao/cft_agr_b9.bps.sas
/home/destr/appli/travail/groupes_de_travail/gw_cft/libre/cft_save/v100/prog/bao/cft_agregation.sas
/home/destr/appli/travail/groupes_de_travail/gw_cft/libre/cft_save/v100/prog/bao/cft_agregation_tittran.bps.sas
```

*Figure 72 : Exemple de récupération des fichiers à traiter*

Le résultat de cette recherche est stocké dans un fichier texte « list_pgm.txt ». Je l'importe en SAS pour récupérer la liste des fichiers à analyser (cf. Figure 73).

```
/* Liste des fichiers : création d'un fichier contenant les programmes puis import */
data _null_;
    call system ("find &rep_in -type f | xargs ls -aR > &rep_out./list_pgm.txt");
run;

data list_pgm;
    infile "&rep_out./list_pgm.txt" delimiter=' ' missover lrecl=32767;
    format  chem_fic $300. chemin $250. fichier $50. ;
    input   chem_fic $ ;
    /* Récupération extension */
    lg_fic = length(chem_fic);
    ext_fic = substr(chem_fic,lg_fic-3,4);
    /* conservation uniquement des .sas */
    if ext_fic = ".sas";

    /* Séparation chemin  - fichier */
    fic_ext = scan(chem_fic,countw(chem_fic,"/"),"/");
    fichier = substr(fic_ext,1,length(fic_ext)-4);
    chemin = substr(chem_fic,1,length(chem_fic)-(length(fic_ext))-1);
run;

data _null_;
    set list_pgm;
    /* répertoire à traiter (2ème paramètre du shell) */
    call symputx("rep_"||left(_N_),chemin);
    /* fichier à traiter (3ème paramètre du shell) */
    call symputx("fic_"||left(_N_),fichier);
    /* nombre de fichier à traiter */
    call symputx("nb_fic",_N_);
run;
```

*Figure 73 : Code SAS de récupération des fichiers à traiter pour l'abaque de migration*

La première étape data (data list_pgm) importe le fichier « list_pgm.txt » dans une table SAS (cf. Figure 74). Cette table est utilisée pour créer les conditions nécessaires de lancement en boucle du traitement Shell sur chacun des fichiers.

| | chem_fic | chemin | fichier |
|---|---|---|---|
| 1 | /home/destr/appli... | /home/destr/appli/travail/groupes_de_travail/gw_cft/libre/cft_save/v100/prog/bao | cft_affec_lib.bps |
| 2 | /home/destr/appli... | /home/destr/appli/travail/groupes_de_travail/gw_cft/libre/cft_save/v100/prog/bao | cft_agr_b9.bps |
| 3 | /home/destr/appli... | /home/destr/appli/travail/groupes_de_travail/gw_cft/libre/cft_save/v100/prog/bao | cft_agregation.bps |
| 4 | /home/destr/appli... | /home/destr/appli/travail/groupes_de_travail/gw_cft/libre/cft_save/v100/prog/bao | cft_agregation_tittran.bps |
| 5 | /home/destr/appli... | /home/destr/appli/travail/groupes_de_travail/gw_cft/libre/cft_save/v100/prog/bao | cft_agr_grp_instr.bps |
| 6 | /home/destr/appli... | /home/destr/appli/travail/groupes_de_travail/gw_cft/libre/cft_save/v100/prog/bao | cft_agr_grp_sect_ctrpt.bps |
| 7 | /home/destr/appli... | /home/destr/appli/travail/groupes_de_travail/gw_cft/libre/cft_save/v100/prog/bao | cft_agr_grp_sect_ref.bps |

*Figure 74 : Exemple de la table SAS des fichiers à traiter par l'outil d'estimation*

La seconde étape (data _null_), en utilisant la table résultat précédente, récupère en macro-variable SAS les répertoires et le nom des programmes à analyser. Il y aura autant de macro-variables que de fichiers à analyser. Les deux macro-variables créés par fichier sont :

- Rep_i : Répertoire du « fichier i » à analyser ;
- Fic_i : Nom du « fichier i » à analyser.

Je stocke le nombre de fichiers à traiter dans la macro-variable « nb_fic ». Cette macro-variable permet de lancer en boucle le traitement Shell qui analyse le contenu d'un fichier.

### 7.2.2 Analyse du contenu des fichiers

Les programmes à analyser sont « retravaillés » avec du code Shell. En effet, dans un premier temps le nombre de lignes était un indicateur. En raisonnant par l'absurde, si une personne programme sur une unique ligne de code, alors l'indicateur du nombre de ligne est faussé car il vaudra un. Le fichier est donc modifié de sorte qu'à chaque « point-virgule » (un point-virgule spécifie une instruction SAS), il y ait un retour chariot. Ainsi chaque ligne correspond, au maximum, à une instruction SAS. Les commentaires sont également supprimés afin de ne pas surévaluer l'indicateur du nombre de lignes. La commande LINUX « SED » soit "éditeur de flux" (Stream EDitor) est largement utilisée pour cela. Sed 's indique un remplacement de texte avec la syntaxe suivante :

s/TexteARemplacer/TexteDeRemplacement/. L'ajout de l'option « g » spécifie de réaliser cette opération sur l'ensemble des occurrences. Par défaut, c'est sur la première rencontrée.

Après cette première étape, les itérations de chaque indicateur sont calculées.

Il y a deux fichiers en sortie du traitement. Le premier nommé « shell_resume_nom_du_fichier_analyse.log » indique le nombre d'occurrence de chaque indicateur dont un exemple est ci-dessous (cf. Figure 75).

```
REP_SAS;PGM_SAS;NB_LIGNE;NB_POINT_VIRGULE;NB_DATA;NB_DATA_DO;NB_PROC;NB_MACRO;NB_MACRO_DO;NB_FUNC
/home/destr/appli/travail/poles_utilisateurs/laser/libre/sas_r;cimport.sas;98;92;4;4;8;5;4;61
```

*Figure 75 : Exemple de fichier resume.log de l'abaque de migration*

L'explication du contenu est le suivant :

- REP_SAS : le répertoire contenant le fichier analysé ;
- PGM_SAS : le nom du fichier analysé ;
- NB_LIGNE : le nombre de ligne du fichier (hors commentaires, hors ligne blanche et avec au maximum une instruction SAS par ligne) ;
- NB_POINT_VIRGULE : le nombre de « ; » qui correspond de façon très proche au nombre d'instruction SAS ;
- NB_DATA : le nombre d'étape « data » ;

- NB_DATA_DO : Le nombre de boucle « DO » ;
- NB_PROC : le nombre de procédure SAS ;
- NB_MACRO : le nombre de macro SAS (cet indicateur sera abandonné très tôt pour manque de pertinence) ;
- NB_MACRO_DO : le nombre de boucle « DO » en macro langage qui peut donc générer beaucoup de code ;
- NB_FUNC : le nombre de fonction.

Le second fichier nommé « shell_func_ nom_du_fichier_analyse.log » contient l'ensemble des fonctions utilisées (cf. Figure 76). Une fonction apparaît dans le fichier résultat autant de fois qu'elle est utilisée dans le programme analysé. Dans l'exemple ci-dessous, la fonction SUBSTR est utilisée quatre fois et la fonction SYMPUTX est utilisée cinq fois.

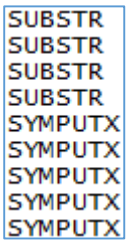


*Figure 76 : Exemple de contenu du fichier fonction de l'abaque de migration*

Pour exécuter le Shell depuis SAS, je me positionne sur le répertoire contenant le programme Shell avec la commande CD (Change Dir) et exécute le Shell sas2r.sh qui nécessite trois paramètres :

- Le répertoire de sortie des résultats
- Le répertoire contenant le fichier à analyser
- Le nom du fichier à analyser

Ces trois paramètres doivent être présents pour l'appel du programme (cf. Figure 78). À l'aide du macro langage SAS et de la boucle DO, j'exécute sur chacun des fichiers le Shell d'analyse.

```
/* boucle sur chaque fichier à analyser */
%do num_fic = 1 %to &nb_fic;

    /* Execution du shell */
    x "cd &rep_shl ; ./sas2r.sh &rep_out &&rep_&num_fic &&fic_&num_fic" ;
```

*Figure 77 : Code SAS d'exécution en boucle du programme Shell d'analyse*

Une fois le Shell exécuté, j'importe et traite les deux fichiers résultats en SAS.

### 7.2.3 Importation des résultats d'analyse et calcul de l'estimation

Le premier fichier résultat est importé avec la PROC IMPORT de SAS (cf. Figure 78).

Le fichier « shell_resume_xxx.log » contient le nombre d'occurrences des indicateurs. Pour établir la note d'un indicateur, il suffit uniquement de multiplier son nombre d'occurrences par la pondération associée.

```
/* boucle sur chaque fichier à analyser */
%do num_fic = 1 %to &nb_fic;

    /* Execution du shell */
    x "cd &rep_shl ; ./sas2r.sh &rep_out &&rep_&num_fic &&fic_&num_fic" ;

    /* Importation du fichier comptant les indicateurs */
    proc import datafile="&rep_out./shell_resume_&&fic_&num_fic...log"
                dbms=dlm
                out=resume_&num_fic(drop=REP_OUT NB_MACRO NB_FUNC
                                    rename=(Nb_point_virgule = nb_instruction))
                replace;
                delimiter=";";
                getnames=yes;
    run;

    /* traitement du fichier d indicateur */
    data resume_&num_fic;
        length pgm_sas $100 rep_sas $250;
        format pgm_sas $100. rep_sas $250.;
        set resume_&num_fic;
        /* calcul de la note par indicateur */
        Note_ligne       = Nb_ligne       * 0.1;
        Note_instruction = Nb_instruction * 0.1;
        Note_data        = Nb_data        * 1;
        Note_proc        = Nb_proc        * 1;
        Note_data_do     = Nb_data_do     * 2;
        Note_macro_do    = Nb_macro_do    * 2;
    run;
```

*Figure 78: Appel du Shell de l'abaque de migration et import des résultats en table SAS*

Le résultat est une table SAS par fichier (cf. Figure 79).

| | pgm_sas | rep_sas | NB_LIGNE | nb_instruction | NB_DATA | NB_DATA_DO | NB_PROC | NB_MACRO_DO | Note_ligne | Note_instruction |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Programme.sas | /home/destr/appli... | 18 | 18 | 2 | 1 | 0 | 1 | 1.8 | 1.8 |

*Figure 79 : Exemple du résultat du premier import du fichier resume du shell*

Il faut désormais traiter le ficher contenant la liste des fonctions afin de calculer l'indicateur associé. Pour rappel, l'indicateur sur les fonctions est calculé avec la somme de racines carrées du nombre d'occurrences des fonctions. J'importe le fichier résultat à l'aide la fonction INFILE.

Ensuite, une première requête SQL compte le nombre d'itérations de chaque fonction. Une deuxième requête somme la racine carrée de chaque nombre d'itération de fonction. Cela donne l'indicateur sur les fonctions. La dernière étape SQL regroupe l'ensemble des indicateurs dans une même table. (cf. Figure 80). Vu qu'il n'y a qu'une ligne par table, il n'y a pas besoin d'établir des conditions de jointures. Ces étapes étant appelées en boucle sur chacun des fichiers, je réalise un PROC APPEND qui cumule les résultats dans une même table SAS. Un exemple détaillé est expliqué juste après ce code.

```
/* Importation du fichier de fonction */
data func_&num_fic;
    length Fonction $50;
    infile "&rep_out./shell_func_&&fic_&num_fic...log";
    input Fonction $ ;
run;
proc sql;
    /* calcul du nombre d'occurence de chaque fonction */
    create table nb_fonc_&num_fic. as
        select count(*) as nb_fct ,
               fonction
        from func_&num_fic.
        group by fonction;
    /* somme de l'ensemble des racine carre des fonctions ==> note fonction */
    create table note_fonc_&num_fic. as
        select count (*) as Nb_Func_dist,
               sum(nb_fct) as Nb_Func,
               sum(sqrt(nb_fct)) as Note_Func
        from nb_fonc_&num_fic.;
    /* jointure entre les indicateurs et la note de fonction */
    create table pond_&num_fic as
        select t1.*,
        t2.Nb_Func_dist,
        t2.Nb_Func,
        t2.Note_Func
        from resume_&num_fic. as t1,
             note_fonc_&num_fic. as t2;
quit;
/* concatenation des resultats de chaque fichier */
proc append base=res analyse data=pond &num fic force;run;
```

*Figure 80 : Code SAS pour calculer l'indicateur sur les fonctions*

La première requête SQL calcule le nombre d'itérations des fonctions (cf. Figure 81).

| nb_fct | Fonction |
|---|---|
| 1 | CATX |
| 5 | DATEPART |
| 1 | EXIST |
| 1 | MAX |

*Figure 81 : Exemple de résultat sur le calcul du nombre d'occurrence des fonctions*

Pour chaque « nb_fct », j'applique la fonction racine carré puis je somme l'intégralité des résultats. Cela donne la note de l'indicateur de fonction (cf. Figure 82).

| | Nb_Func_dist | Nb_Func | Note_Func |
|---|---|---|---|
| 1 | 18 | 127 | 36.223204316 |

*Figure 82 : Exemple de résultat sur l'indicateur des fonctions*

L'ensemble des indicateurs est calculé, il est désormais nécessaire de les mettre en forme pour les présenter.

### 7.2.4 Mise en forme des résultats d'estimation

Afin que le résultat soit le plus compréhensible possible j'affiche clairement le détail du score de complexité.

L'étape « data » ci-dessous (cf. Figure 83) calcule :

- La somme de l'indicateur des boucles « do » (indicateur de boucle do dans une étape data + indicateur de boucle do en macro langage) ;
- La note de complexité totale en faisant la somme de l'ensemble des notes des indicateurs ;
    - Note_old : La première version de la note de complexité avec le nombre de ligne ;
    - Note : La version actuelle de la note de complexité avec le nombre d'instruction SAS.
- L'estimation de charge de migration en multipliant la note de complexité par 0,091 (corrélation entre le score et la charge) ;
- D'afficher les intervalles d'estimation de charge et de complexité.

```
/* creation des formats pour le score */
proc format;
    value classe
        0-22      = 1
        22<-55    = 2
        55<-110   = 3
        110<-165  = 4
        165<-high = 5;
    value intrvl
        0-22      = "[1 - 2]"
        22<-55    = "[2 - 5]"
        55<-110   = "[5 - 10]"
        110<-165  = "[10 - 15]"
        165<-high = "[15+]";
run;

data &tab_sortie;
    set res_analyse;
        length Intrvl_jours $10;
        Note_Boucle  = sum(note_data_do, note_macro_do);
        Note_old     = sum(Note_Ligne,Note_Data,Note_Proc,Note_Func, Note_Boucle);
        Note         = sum(Note_Instruction,Note_Data,Note_Proc,Note_Func, Note_Boucle);
        Intrvl_jours = put(Note, intrvl.);
        /* la correspondance entre la note et l estimation de jour est de 0,091 */
        Estim_old    = Note_old * 0.091;
        Estim_jours = Note     * 0.091;
        Classe       = put(Note, classe.);
run;
```

*Figure 83 : Création d'un abaque de migration - Code SAS de mise en forme du résultat final*

Le résultat est alors une table SAS avec le format suivant (cf. Figure 84) :

| | rep_sas | pgm_sas | note_instruction | note_data | note_proc | note_func | note | estim_jours | Note_Boucle | intrvl_jours | classe |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | total | nombre de pgm : 85 | 1361.1 | 837 | 816 | 2712.7511675 | 6294.8511675 | 572.83145624 | 568 | [15+] | 5 |
| 2 | /home/destr... | clc_3.bps.sas | 122.3 | 159 | 82 | 120.75507566 | 592.05507566 | 53.877011885 | 108 | [15+] | 5 |
| 3 | /home/destr... | clc_1.bps.sas | 83.4 | 82 | 49 | 116.67702183 | 383.07702183 | 34.860008987 | 52 | [15+] | 5 |
| 4 | /home/destr... | clc_0.bps.sas | 81.7 | 93 | 51 | 117.1610652 | 352.8610652 | 32.110356933 | 10 | [15+] | 5 |
| 5 | /home/destr... | ros_extract.bps.sas | 80.4 | 25 | 64 | 140.86126094 | 318.26126094 | 28.961774746 | 8 | [15+] | 5 |
| 6 | /home/destr... | ros_extract.sas | 80.4 | 25 | 64 | 140.86126094 | 318.26126094 | 28.961774746 | 8 | [15+] | 5 |
| 7 | /home/destr... | cft_aga.bps.sas | 52.3 | 37 | 32 | 103.33097626 | 268.63097626 | 24.445418839 | 44 | [15+] | 5 |
| 8 | /home/destr... | cft_rcf.bps.sas | 54.2 | 28 | 21 | 64.330913287 | 201.53091329 | 18.339313109 | 34 | [15+] | 5 |
| 9 | /home/destr... | cft_choix_trt.sas | 47.2 | 12 | 11 | 68.918291328 | 179.11829133 | 16.299764511 | 40 | [15+] | 5 |
| 10 | /home/destr... | cft_rcv.bps.sas | 38 | 18 | 36 | 72.653929192 | 178.65392919 | 16.257507556 | 14 | [15+] | 5 |
| 11 | /home/destr... | ros_extract_exo.bps... | 28.8 | 14 | 14 | 78.165884688 | 142.96588469 | 13.009895507 | 8 | [10 - 15] | 4 |
| 12 | /home/destr... | ros_extract_exo.sas | 28.8 | 14 | 14 | 78.165884688 | 142.96588469 | 13.009895507 | 8 | [10 - 15] | 4 |
| 13 | /home/destr... | cft_tp_sesof_sdmx.b... | 48.9 | 2 | 0 | 19.569707984 | 130.46970798 | 11.872743427 | 60 | [10 - 15] | 4 |

*Figure 84 : Résultat en table SAS de l'outil d'abaque*

Avec une présentation Excel soignée (cf. Tableau 6), on identifie facilement le coût total de migration d'un projet (comme MIR ou CFT), le détail par programme et l'ensemble des indicateurs.

| Rep_SAS | Pgm_SAS | Intrvl_jours | Estim_jours | Classe | Note | NOTE_LIGNE | NOTE_DATA | NOTE_PROC | NOTE_FUNC | NOTE_BOUCLE |
|---|---|---|---|---|---|---|---|---|---|---|
| Total | Nombre de pgm : 59 | [15+] | 264 | 5 | 2 898 | 848 | 404 | 364 | 1 117 | 165 |
| MIR | mir_red.bps.sas | [15+] | 18 | 5 | 203 | 50 | 52 | 31 | 50 | 21 |
| | mir_supermain_CIM.sas | [10 - 15] | 11 | 4 | 116 | 29 | 13 | 16 | 56 | 3 |
| | mir_choix_trt.sas | [10 - 15] | 10 | 4 | 113 | 27 | 4 | 7 | 18 | 57 |
| | mir_connectique_pgm_sse.bps.s | [10 - 15] | 10 | 4 | 112 | 26 | 23 | 28 | 32 | 3 |
| | mir_creation_table_cor_kmanue | [10 - 15] | 10 | 4 | 111 | 47 | 13 | 20 | 31 | 0 |
| | mir_agr.bps.sas | [5 - 10] | 10 | 3 | 110 | 31 | 16 | 17 | 39 | 6 |
| | mir_cma.bps.sas | [5 - 10] | 9 | 3 | 97 | 29 | 10 | 16 | 30 | 12 |
| | mir_cmi.bps.sas | [5 - 10] | 9 | 3 | 95 | 32 | 10 | 13 | 25 | 15 |
| | mir_bce.bps.sas | [5 - 10] | 8 | 3 | 92 | 26 | 13 | 10 | 44 | 0 |
| | mir_ags.bps.sas | [5 - 10] | 8 | 3 | 91 | 30 | 20 | 14 | 26 | 0 |
| | mir_txa.bps.sas | [5 - 10] | 8 | 3 | 90 | 31 | 15 | 9 | 35 | 0 |
| | mir_agg.bps.sas | [2 - 5] | 5 | 2 | 53 | 18 | 6 | 9 | 20 | 0 |
| | mir_rab.bps.sas | [2 - 5] | 4 | 2 | 44 | 14 | 7 | 5 | 18 | 0 |
| | mir_lst_demprec.sas | [2 - 5] | 4 | 2 | 43 | 13 | 3 | 11 | 16 | 0 |
| | mir_charge_param_ros.bps.sas | [1 - 2] | 1 | 1 | 15 | 5 | 0 | 3 | 7 | 0 |
| | mir_extract_corr.bps.sas | [1 - 2] | 1 | 1 | 10 | 3 | 0 | 2 | 5 | 0 |
| Total | Nombre de pgm : 81 | [15+] | 512 | 5 | 5 623 | 1 401 | 791 | 665 | 1 984 | 783 |
| CFT | clc_3.bps.sas | [15+] | 58 | 5 | 637 | 130 | 163 | 81 | 104 | 159 |
| | ros_extract.sas | [15+] | 31 | 5 | 336 | 113 | 25 | 64 | 122 | 12 |
| | cft_aga.bps.sas | [15+] | 24 | 5 | 259 | 63 | 36 | 28 | 72 | 60 |
| | cft_rcf.bps.sas | [15+] | 20 | 5 | 220 | 57 | 28 | 20 | 64 | 51 |
| | cft_tp_sesof_sdmx.bps.sas | [10 - 15] | 15 | 4 | 160 | 50 | 2 | 0 | 19 | 90 |
| | cft_rcv.bps.sas | [10 - 15] | 14 | 4 | 158 | 43 | 16 | 26 | 57 | 15 |
| | clc_4.bps.sas | [5 - 10] | 9 | 3 | 104 | 26 | 32 | 12 | 34 | 0 |
| | cft_env_processus.bps.sas | [5 - 10] | 9 | 3 | 99 | 24 | 11 | 10 | 44 | 9 |
| | cft_cma.bps.sas | [5 - 10] | 9 | 3 | 94 | 21 | 13 | 10 | 51 | 0 |
| /home/dest | Import_XML.sas | [10 - 15] | 11 | 4 | 119 | 38 | 20 | 22 | 36 | 3 |

*Tableau 6 : Exemple de résultat de l'abaque de migration SAS vers R*

Ce formalisme et les estimations ont satisfait pleinement la direction de la DGS.

## 7.3 LES LIMITES DE L'OUTIL D'ESTIMATION

L'outil permet de façon automatique de scanner le contenu d'un répertoire et d'identifier correctement la complexité d'un programme. Son estimation de charge de migration, semble correcte en moyenne, mais à cette date, je manque de recul pour le vérifier, tout en ayant déjà identifié certaines limites de l'outil.

Le paramètre humain du niveau de la (ou des) personne réalisant la migration de SAS vers R n'est pas pris en compte. Une personne experte SAS et R mettra normalement moins de temps à migrer un traitement qu'une personne maitrisant moins bien voire peu une ou les deux technologies. Il en va de même si le programme est « bien codé » et bien commenté. Dans ces cas, la migration mettra moins de temps que si le programme est inutilement complexe et sans commentaire.

L'outil surestime les codes à « répétition ». Par exemple, des successions de « else if » avec uniquement une valeur qui change sera surévaluée. Il faudrait pouvoir réaliser une analyse textuelle pour le corriger. Mais cela deviendrait beaucoup plus complexe et plus coûteux à mettre en œuvre. De plus l'objectif n'est pas d'avoir un chiffre « exact » sur un programme mais bien une idée globale.

SAS dans la version Entreprise Guide permet de créer des étapes en « clic bouton » et d'insérer des programmes uniquement stockées dans les métadonnées. L'outil ne l'estime pas automatiquement. Il faut copier ces codes dans des programmes pour qu'il puisse les estimer.

La reprise de l'historique des données n'est pas à négliger. Il faut, bien entendu, écrire les nouveaux programmes en R mais il faut également conserver les données historiques (si nécessaire). A la DGS, beaucoup d'études et de productions stockent uniquement leurs résultats en tables SAS. Il faudra alors, soit constituer des datamart externes (SQL SERVER, Oracle…) ou soit convertir l'historique des données SAS en R. Le choix de la solution devra se faire au cas par cas. L'outil n'estime pas cette partie, qui est donc un coût supplémentaire.

De nombreux programmes sont également sur Windows mais l'abaque ne fonctionne pas sur cette plateforme. En effet les commandes Shell utilisées ne sont pas présentes sur Windows. Je suis donc en train d'étudier l'utilisation de la plateforme Cygwin qui permettrait d'utiliser du Shell sur du Windows.

Une autre problématique est soulevée sur l'identification des traitements à migrer en priorité. J'ai été en charge de les identifier. Il y a bien entendu la priorisation « métier » mais j'ai cherché à connaître de façon « informatique » les traitements les plus utilisés et donc potentiellement prioritaires.

## 7.4 IDENTIFICATION DU PARC DE MIGRATION

La DGS utilisant un serveur LINUX pour le logiciel SAS, il est possible de récupérer les dates de modification et de dernier accès à un fichier avec la commande « stat ». N'étant pas expert du langage Shell et afin de gagner du temps j'ai demandé de l'aide à l'OI. Ils m'ont

livré un script parcourant l'intégralité d'un répertoire de façon récursive et qui récupère les informations de date dans un fichier csv (cf. Figure 85).

```
#/usr/bin/ksh

find $1 -name "*.sas" -exec stat {} + > $HOME/statsas.lst

cat  $HOME/statsas.lst | awk '  BEGIN { FS=":"} { if ( $1==  File" ) { printf $2 ";"  }
                                else  if ( $1=="Access" && index($2,"(" ) != 0 ) { nbc=split($2,a,"(") ;  nbb=split($3,b,"(") ;  nba=split($4,c,"(") ;
                                printf substr(a[2],1,15)  ";" substr(b[2],1,index(b[2],")")-1)  ":"  substr(c[2],1,index(c[2],")")-1) ";" }
                                else  if  ( $1=="Access" ) { printf substr($2,1,14) ":" $3 ":" substr($4,1,2) ";" }
                                else if ( $1=="Modify" ) { printf substr($2,1,14) ":" $3 ":" substr($4,1,2) ";" }
                                else  if ( $1=="Change" ) { printf substr($2,1,14) ":" $3 ":" substr($4,1,2) ";\n" } } ' > $HOME/statsas.csv
```

*Figure 85 : Programme Shell récupérant les dates d'accès, de modification du programme et des modifications des droits*

En SAS, j'importe le fichier csv « statsas.csv » afin de pouvoir réaliser une distribution sur la date de dernier accès et une sur la date d'enregistrement du fichier (cf. Figure 86 et Figure 87).

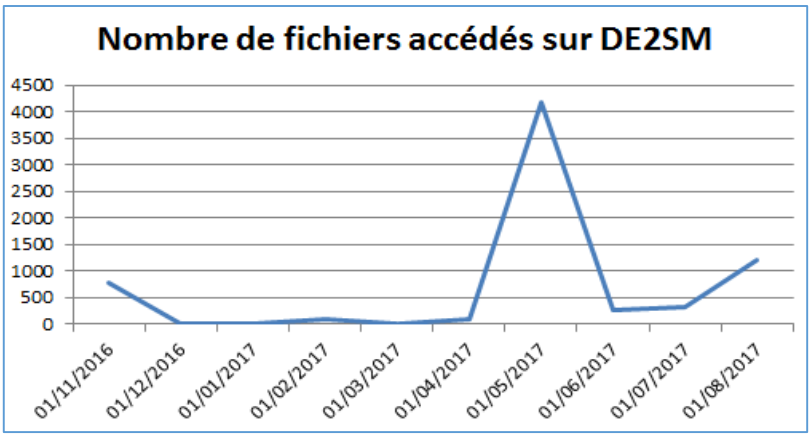


*Figure 86 : Distribution des dates d'accès aux programmes SAS sur le serveur LINUX*

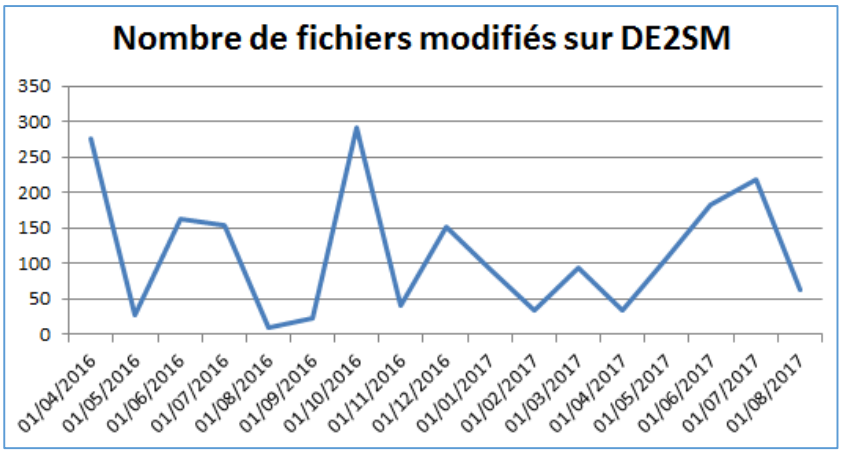


*Figure 87 : Distribution des dates d'enregistrement aux programmes SAS sur le serveur LINUX*

Je me suis rendu compte que la dernière date d'accès n'est pas exacte. En effet, l'information contenue est le premier accès depuis la dernière sauvegarde. Après ce premier

accès la date n'est plus mise à jour. L'option permettant de mettre à jour cette métadonnée à chaque accès de fichier n'est pas activée sur le serveur LINUX de la DGS.

De plus, deux évènements importants ont modifié l'information d'accès aux fichiers. En novembre 2016, la DGS est passée d'un serveur UNIX à LINUX. C'est pourquoi la date la plus ancienne d'accès remonte à ce mois. Le 12/05/2017, une réorganisation de file system a mis à jour la date d'accès de l'ensemble du répertoire « poles_utilisateurs », soit environ 4170 programmes, ce qui explique le pic sur le graphique.

Environ une centaine de programmes sont modifiés chaque mois. Cela peut être uniquement un paramètre pour changer l'échéance d'une production.

Avec la contrainte sur la mise à jour de la dernière date d'accès, il est difficile d'établir « informatiquement » les programmes à migrer. La priorisation a donc été « humaine » en choisissant les programmes récurrents ou ceux dont l'obsolescence fonctionnelle était proche.

J'ai donc abordé l'aspect « plus technique » de la migration, mais sans oublier l'aspect « humain ». En effet même si une migration d'un logiciel vers un autre est faisable à 100%, cela ne signifie pas que les personnes en place y arriveront aisément. Il y a nécessité de les accompagner. Outre les formations sur R et Python déjà initiées, j'ai proposé et ensuite géré un accompagnement sur l'utilisation de R (et à terme également Python).

## 7.5 LA CONDUITE DU CHANGEMENT

Le succès d'une conduite du changement, et donc dans notre cas le succès de la migration ne se fera pas sans une adhésion des utilisateurs. Il existe de multiples résistances au changement (peur, remise en cause des compétences…) [20]. Il y a donc une nécessité d'accompagnement au changement. J'ai la charge d'accompagner ce changement auprès des utilisateurs de la DGS.

J'ai initié un suivi mensuel de l'utilisation de R, en y conviant également des experts R de la Banque de France ainsi que les développeurs R de la DGS, sur le base du volontariat. Je sollicite, afin d'enrichir les connaissances de chacun, un expert pour une présentation d'une fonction, d'un package, etc. La dernière présentation était sur les package Tidyverse

(https://www.tidyverse.org/). L'important est que les utilisateurs ne se sentent pas « abandonnés » devant un nouveau logiciel juste avec quelques jours de formations.

J'ai également initié un guide utilisateur R pour donner des conseils sur R (cf. Figure 88). Je l'alimente au fur et à mesure des questions récurrentes des utilisateurs.

Si vous voulez accéder à des données SAS, il faut alors y déposer sur le répertoire de partage vos données SAS comme l'exemple ci-dessous.

```
/* repertoire de partage entre R et SAS */
libname ech_r '/home/destr/appli/travail/echange_r';
data ech_r.test_ech;
     var_num    = 10;
     var2_car   = 'abc';
     var_date_j = '25sep2017'd;
     var_date_s = '25sep2017 10:25:13'dt;
run;
```

Puis lecture en R :

```
#install.packages("sas7bdat")
library(sas7bdat)
test <- read.sas7bdat("/home/destr/appli/travail/echange_r/test_ech.sas7bdat")
View(test)
```

Le résultat sera la Dataframe R ci-dessous (Figure 7)

| | var_num | var2_car | var_date_j | var_date_s |
|---|---|---|---|---|
| 1 | 10 | abc | 21087 | 1821916800 |

Figure 7 : Dataframe exemple de lecture de données SAS

Pour spécifier que var_date_j est une date en nombre de jour dont l'origine est le 01/01/1960 (origine des dates SAS) et en ajouter un nouveau vecteur il faut écrire :

```
#création d'un nouveau vecteur var_date_j2 depuis une date jour
test$var_date_j2 <- as.Date(test$var_date_j, origin = "1960-01-01")
#spécifier que var_date_j est une date sans création d'un vecteur
test<- as.Date(test$var_date_j, origin = "1960-01-01")

#création d'un vecteur var_date_s2 depuis une date en jour-seconde
test$var_date_s2<-as.POSIXct(test$var_date_s,origin="1960-01-01 00:00:00", tz="UTC")
```

Le résultat avec la première et la dernière ligne de code est le suivant (cf. Figure 8) :

| | var_num | var2_car | var_date_j | var_date_s | var_date_j2 | var_date_s2 |
|---|---|---|---|---|---|---|
| 1 | 10 | abc | 21087 | 1821954313 | 2017-09-25 | 2017-09-25 10:25:13 |

Figure 8 : Dataframe exemple avec création de deux variables date

*Figure 88 : Extrait du guide utilisateur R sur la lecture de table SAS*

Il est nécessaire dégager du temps pour la migration. Mais réaliser les productions courantes et répondre aux nouveaux besoins restent la priorité. Il y a donc une nécessité, en plus de ces tâches, de donner du temps aux personnes pour migrer les traitements ou même pour répondre à un nouveau besoin mais cette fois-ci en R et non en SAS.

J’ai conseillé de nommer dans chaque service une personne « référente » pour la migration. Avoir une personne à proximité, qui plus est sur le même domaine fonctionnel, rassure.

La DGS a fait le choix d’une migration progressive qui laisse le temps de monter en compétence. Il n’a pas été fait le choix d’embaucher en masse des prestataires de services pour migrer rapidement une grande partie des traitements. Par contre dans les équipes où il y avait des prestataires de service en SAS, il a été demandé d’avoir également des compétences sur R. C’est le cas de mon équipe.

# 8 CONCLUSION

La migration partielle de la chaîne de calcul du taux d'usure s'est effectuée avec succès. La solution mise en place répond aux besoins et exigences métier. Les données sont correctement transmises et stockées dans le nouvel système d'information. Les résultats de la chaîne de calcul SAS sont conformes. Un bon cadrage de début de projet a permis de cibler les évolutions dans les spécifications techniques et dans le code SAS. Le passage en table de paramètre des caractéristiques des taux d'usure permet d'obtenir désormais leur historique d'évolution et, de plus, une possibilité de rejouer une période précédente facilement. Une réflexion plus approfondie sur le stockage des données dans la base ROSTAM a optimisé les extractions de l'ensemble des traitements utilisant les données de la table des faits. Mes objectifs, en tant que chef de projet, ont donc été atteints.

Néanmoins, après cette expérience et un certain recul j'ai fait deux constats. L'utilisation de EAI pour transmettre les données résultats de SAS vers ROSTAM a été un centre de coût inutile. L'utilisation de la fonction « load » d'un ETL aurait économisée du temps de développement et également une simplification de la chaîne de production. Le modèle en étoile pour la base de données ROSTAM n'a pas été correctement implémenté. Il en résulte une compréhension compliquée du modèle mais également, à terme, des temps d'exécution de requête plus grands. Cela montre la difficulté de conceptualiser un modèle de données correct et optimisé. Je regrette de ne pas avoir réussi à convaincre les interlocuteurs en charge de cette tâche et de ne pas avoir participé davantage à sa conception.

Ma seconde problématique était de proposer une méthodologie de migration de SAS vers R. J'ai alors conceptualisé et réalisé un outil estimant la charge et la complexité de migration de SAS vers R d'un ou plusieurs programmes SAS. En effet, obtenir une approximation du coût de développement est un point important dans une migration. Le postulat de base est, à niveau de compétence égale, tout programme SAS met autant de temps à être développé en R. Via un programme en langage Shell et SAS, j'analyse les programmes sur le nombre d'instructions SAS, d'étape data, de procédure SAS, de boucle « do » et nombre de fonctions appelées. Cela établit un score, une complexité et une estimation de temps de migration. Ces informations sont retranscrites dans une table SAS. Cet outil permet de parcourir l'intégralité du contenu d'un répertoire (et donc de ces sous-répertoires). Les résultats ont été satisfaisants mais avec certaines limites. En effet, La redondance de code, par exemple

avec de multiples « else if » est surévaluée. Un œil « humain » est donc nécessaire pour, le cas échéant, diminuer la charge estimée. Il manque, à date de la rédaction, du recul pour vérifier la pertinence des estimations de l'outil. Cet outil répond à la problématique posée par ma direction, très satisfaite des résultats.

Pour les utilisateurs, la migration de SAS vers R est très impactante. Il est nécessaire d'appliquer une conduite de changement. Dans cette optique, j'ai procédé notamment à la mise en place de trois outils :

- Création d'un guide utilisateurs avec des exemples simples de programmation R comme l'extraction et la manipulation de données ;
- Mise à jour régulière de ce guide ;
- Animation des comités de suivi mensuel de migration R.

Par ailleurs, une communauté R interne DGS est également créée pour échanger programmes et conseils.

Ce projet se caractérisait par sa complexité technique et humaine ainsi que par son envergure. J'en ai retiré un double enseignement :

- Une forte réflexion sur la conception, lors du lancement, est une des clés de la réussite du projet ;
- Aussi savoir questionner ses choix et la réalisation envisagée pour vérifier s'il n'y a pas de meilleure méthode à appliquer.

Tout au long de ce travail, les inévitables difficultés rencontrées, techniques et humaines, m'ont permis d'évoluer et d'améliorer ma communication avec tous les acteurs ; qu'ils en soient ici remerciés.

J'ai également apprécié d'avoir une problématique double. Une où il s'agissait d'analyser l'existant pour le faire évoluer avec une collaboration inter-équipes. Et une autre où je devais personnellement conceptualiser une idée et la mettre en œuvre à partir de zéro. Ce sont deux méthodes complétement différentes et je suis satisfait d'avoir répondu aux attentes et problématiques dans les deux cas.

Actuellement, je suis le chef de projet en charge de migrer la chaîne statistique des BSI (Balance Sheet Items) de SAS vers R. Ce projet contient en plus une refonte fonctionnelle

importante. Son coût de migration est estimé à 450jh. Trois développeurs sont à temps pleins sur ce projet. Je présente également à d'autres directions de la Banque de France l'outil d'estimation de migration et je délivre des conseils sur la mise en place de ces projets. L'année 2018 se profile donc dans la continuité de mon projet de mémoire.

# BIBLIOGRAPHIE


[1] H. Verjus, Conception et construction de fédérations de progiciels, HAL Id: tel-00010877, Thèse de Doctorat: Université de Savoie, 2005, p. 264.

[2] J. Fenner, «Enterprise Application Integration Techniques, http://www0.cs.ucl.ac.uk/staff/ucacwxe/lectures/3C05-02-03/aswe21-essay.pdf».

[3] C. Madeleine, «Enterprise Application Integration : EAI,» *Techniques de l'ingénieur - H2470,* 2004.

[4] L. Stumpf, «Enterprise Application Integration,» 2006. [En ligne]. Available: deptinfo.cnam.fr/new/spip.php?pdoc681. [Accès le 18 07 2017].

[5] F. Rivard, «EAI De l'intégration à l'e-business, livre blanc,» 2000.

[6] D. Rousse, «Panorama d'une infrastructure EAI,» Centre national de la recherche scientifique, Direction des systèmes d'information, 2003. [En ligne]. Available: http://david.rousse.free.fr/download/cnrs_dsi/Panorama_EAI.pdf. [Accès le 19 07 2017].

[7] B. ESPINASSE, Cours Stratégies de développement des Systèmes d'Information Opérationnels de l'entreprise, Ecole Polytechnique Universitaire de Marseille, 2014.

[8] B. Burquier, Business Intelligence avec SQL SERVER 2008 : Mise en œuvre d'un projet décisionnel, Dunod, 2009, p. 432.

[9] R. Kimball, M. Ross, W. Thornthwaite, J. Mundy et B. Becker, The Data Warehouse Lifecycle Toolkit, Wiley, 2008, p. 672.

[10] B. Inmon, Building the Data Warehouse, 3e éd., Wiley, 2002, p. 406.

[11] R. Kimball et J. Caserta, The Data Warehouse ETL Toolkit, Wiley, 2004, p. 491.

[12] T. GRUBER, Toward Principles for the Design of Ontologies Used for Knowledge Sharing - International Journal of Human and Computer Studies, 43(5/6): 907-928, 1995.

[13] M. A. Mestiri, Vers une approche web sémantique dans les applications de gestion de conférences, http://theses.ulaval.ca/archimede/fichiers/24629/24629.html, 2007, chapitre 2.

[14] H. Jaudoin, Thèse DEA Informatique : Réécriture de requêtes en termes de vues en présence de contraintes de valeurs pour un système d'intégration de sources de données agricoles , chapitre 1, HAL Id: tel-00684041, 2012.

[15] C. Chrisment, G. Pujolle, F. Ravat, O. Teste et G. Zurfluh, «Entrepôts de données,» *Techniques de l'ingénieur - H3870,* 2005.

[16] D. C. Faye, These Médiation de données sémantique dans SenPeer, un système pair-à-pair de gestion de données, tel-00481311, 2007.

[17] A. Doucet, «Cours "Médiateurs", http://www-poleia.lip6.fr/~doucet/CoursBDIA/Cours5.pdf».

[18] B. Espinasse, cours Ecole Polytechnique Universitaire de Marseille "Entrepôts de données : Systèmes ROLAP, MOLAP et HOLAP", http://www.lsis.org/espinasseb/Supports/DWDM/5-SystemesOLAP-4p.pdf, 2015.

[19] E. Metais, «Encyclopedia Universalis, chapitre "Systèmes d'aide à la décision et entrepôts de données" ISBN 978-2-85229-337-3,» 2010.

[20] T. Rocves, CNAM DSY101, Le changement organisationnel, 2015.

[21] C. Plumejeaud, «Enterprise Application Integration,» 2008. [En ligne]. Available: lig-membres.imag.fr/plumejeaud/NFE107-fichesLecture/EAI.ppt. [Accès le 18 07 2017].

[22] R. Kimball, «http://www.kimballgroup.com/,» [En ligne]. [Accès le 07 08 2017].

[23] B. Willem, Construction of engineering ontologies for knowledge sharing and reuse, 1997, Thèse,ISBN: 90-365-0988-2, pp. 1-23.

[24] «Banque de France,» [En ligne]. Available: https://www.banque-france.fr/. [Accès le 11 07 2017].

# ANNEXE

**Annexe 1 : Les catégories de crédits pour le taux d'usure**

| Catégorie | Libellé |
|---|---|
| IA3 | Prêts d'un montant inférieur ou égal à 3000 € |
| 3A6 | Prêts d'un montant supérieur à 3000 € et inférieur ou égal à 6000 € |
| SU6 | Prêts d'un montant supérieur à 6000 € |
| IMV | Prêts à taux variable |
| IMF1 | Prêts d'une durée inférieure à 10 ans |
| IMF2 | Prêts d'une durée comprise entre 10 ans et moins de 20 ans |
| IMF3 | Prêts d'une durée de 20 ans et plus |
| IMR | Prêts relais |
| VAT1 | Prêts consentis en vue d'achats ou de ventes à tempérament des personnes morales non commerciales |
| VAT2 | Prêts consentis en vue d'achats ou de ventes à tempérament des personnes morales commerciales et entrepreneurs individuels |
| S2V1 | Prêts d'une durée initiale supérieure à 2 ans, à taux variable des personnes morales non commerciales |
| S2V2 | Prêts d'une durée initiale supérieure à 2 ans, à taux variable des personnes morales commerciales et entrepreneurs individuels |
| S2F1 | Prêts d'une durée initiale supérieure à 2 ans, à taux fixe des personnes morales non commerciales |
| S2F2 | Prêts d'une durée initiale supérieure à 2 ans, à taux fixe des personnes morales commerciales et entrepreneurs individuels |
| AI21 | Autres prêts d'une durée initiale inférieure ou égale à 2 ans des personnes morales non commerciales |
| AI22 | Autres prêts d'une durée initiale inférieure ou égale à 2 ans des personnes morales commerciales et entrepreneurs individuels |
| DEC1 | Découverts en compte des personnes morales non commerciales |
| DEC2 | Découverts en compte des personnes morales commerciales et entrepreneurs individuels |

*Tableau 7 : Les catégories de crédit du taux d'usure*

## Annexe 2 : Structure du fichier XML de la collecte M_CONTRAN pour le formulaire MCO1

| CODE XML | LIBELLE | TYPE | LONGUEUR | PRESENCE (OBligatoire / FAcultarif / COnditionnel) | COMMENTAIRES |
|---|---|---|---|---|---|
| **SEQ** | Identifiant OneGAte | Numérique | | OB | Clé OneGate<br>Séquence permettant de classer les différents Items. |
| **ID_GUI** | Code guichet | Alpha | 5 | FA | Le code guichet n'est servi que pour les établissements généralistes.<br>Les établissements spécialisés ne doivent pas renseigner de code guichet.<br>Les établissements généralistes doivent précéder le code guichet d'un nombre de 0 suffisant pour que la longueur de la valeur corresponde à la longueur requise. |
| **RFLICR** | Référence du crédit | Alpha | 14 | OB | Numéro d'ordre du crédit octroyé :<br>numéro séquentiel, indiquant le numéro du crédit considéré tel que fixé par l'établissement |
| **INS_FI** | Catégorie de Nature de l'instrument financier | Numérique | 3 | OB | Les valeurs suivantes doivent être déclarées :<br>100 Découverts<br>200 Escompte et assimilé<br>210 Financement sur Loi Dailly<br>220 Autres créances commerciales<br>230 Mobilisation de créances sur l'étranger<br>240 Crédits fournisseurs<br>250 Crédits commerciaux à des nonrésidents<br>260 Autres crédits à l'export<br>300 Financement de ventes à tempérament<br>310 Prêts personnels<br>320 Crédits revolving ou crédits permanents<br>330 Prêts sur carte de crédit<br>400 Facilités d'émission<br>410 Crédit global d'exploitation<br>420 Financement de stocks<br>430 Avances sur avoirs financiers<br>440 Autres crédits de trésorerie<br>500 Crédits à l'équipement aidés<br>510 Autres crédits à l'équipement<br>600 Crédits à l'habitat non réglementés<br>610 Prêts aux organismes HLM<br>620 PLA<br>630 PLI<br>640 Prêts aidés d'accession à la propriété<br>650 Prêts conventionnés<br>660 Prêts bancaires conventionnés (PBC) |

| CODE XML | LIBELLE | TYPE | LONGUEUR | PRESENCE (OBligatoire / FAcultarif / COnditionnel) | COMMENTAIRES |
|---|---|---|---|---|---|
| | | | | | 670 PEL<br>680 Autres prêts réglementés<br>690 Crédits promoteurs<br>700 Autres crédits à la clientèle<br>800 Prêts subordonnés<br>900 Crédit-bail mobilier<br>910 Crédit-bail immobilier<br>920 Crédit-bail sur actifs incorporels |
| **MT_CRDT** | Montant du crédit | Numérique | 11 | OB | Le montant du concours accordé, exprimé en euros (sans décimale). La valeur est strictement positive. |
| **MT_MAX** | Montant maximum autorisé | Numérique | 11 | CO | Le montant maximum autorisé, exprimé en euros (sans décimale). La valeur est positive ou nulle.<br><br>Le montant maximum autorisé doit être renseigné uniquement pour les découverts, crédits permanents et prêts sur carte de crédit, interdit sinon. Il correspond au montant maximum susceptible d'être mis à la disposition du client au cours du mois de référence. |
| **PRT_POOL** | Part dans le pool | Numérique | 3 | OB | La part dans le pool doit être obligatoirement saisie pour tout crédit déclaré. Elle doit être exprimée en pourcentage sans décimale, être strictement positive et inférieure ou égale à 100. |
| **DUREE_IN** | Durée initiale | Numérique | 3 | CO | La durée initiale de l'opération, renseignée en nombre entier de mois.<br><br>La valeur est strictement positive. |
| **CDT_NGCT** | Conditions de négociation | Numérique | 1 | OB | Cette rubrique devra être codifiée de la façon<br><br>suivante :<br><br>• Autres cas : 0<br>• Cas d'une reconduction tacite : 1<br>• Cas d'un prêt renégocié : 2 |
| **USG_PRT** | Usage du prêt | Numérique | 1 | OB | Il devra être spécifiquement déclaré si l'objet du prêt est lié à l'activité professionnelle de l'entrepreneur individuel ou si le prêt est destiné à faire face à un besoin personnel du ménage de l'emprunteur. Cette rubrique devra être codifiée de la façon suivante :<br><br>• Prêt à usage professionnel : 0<br>• Prêt à usage personnel : 1 |
| **IDX_REF** | Index de référence | Numérique | 1 | CO | L'index de référence doit être codifié de la<br><br>manière suivante :<br><br>- Taux fixe : 0<br><br>- Taux variable indexé sur :<br><br>• TBB : 1<br>• EONIA : 2<br>• EURIBOR 1 mois : 3<br>• EURIBOR 3 mois : 4<br>• EURIBOR 1 an : 5<br>• TMO ou TME : 6<br>• - Autre formule ou mixte : 7 |
| **PFIT** | PFIT | Numérique | 1 | CO | La période de fixation initiale du taux (PFIT) de l'opération, codifiée |

| CODE XML | LIBELLE | TYPE | LONGUEUR | PRESENCE (OBligatoire / FAcultarif / COnditionnel) | COMMENTAIRES |
|---|---|---|---|---|---|
| | | | | | de la manière suivante :<br><br>PFIT ≤ 3 mois : 0<br><br>• 3 mois < PFIT ≤ 1 an : 1<br>• 1 an < PFIT ≤ 3 ans : 2<br>• 3 ans < PFIT ≤ 5 ans : 3<br>• 5 ans < PFIT ≤ 10 ans : 4<br>• 10 ans < PFIT : 5 |
| **TESE** | TESE | Numérique | | OB | Le TESE (Taux Effectif au Sens Etroit) est renseigné sur 6 caractères (4 décimales après la virgule, même s'il s'agit de zéros) et indiqués sans virgule ni point décimal.<br><br>Précéder le TESE d'un nombre de 0 suffisant pour que la longueur de la valeur corresponde à la longueur requise. |
| **TEG** | TEG | Numérique | 6 | OB | Le TEG (Taux Effectif Global) est renseigné sur 6 caractères (4 décimales après la virgule, même s'il s'agit de zéros) et indiqués sans virgule ni point décimal.<br><br>Précéder le TEG d'un nombre de 0 suffisant pour que la longueur de la valeur corresponde à la longueur requise. |
| **CAP** | CAP | Numérique | 6 | CO | Le CAP est renseigné sur 6 caractères (4 décimales après la virgule, même s'il s'agit de zéros) et indiqués sans virgule ni point décimal.<br><br>Pour les crédits à taux variable non plafonné, le CAP a pour valeur 999999.<br><br>Précéder le CAP d'un nombre de 0 suffisant pour que la longueur de la valeur corresponde à la longueur requise. |
| **AJUST** | Mode d'ajustement | Numérique | 1 | CO | Pour les crédits à taux variable, le mode d'ajustement du remboursement du crédit prévu dans les conditions du contrat est codifié de la manière suivante :<br><br>• Ajustement par la durée : 0<br>• Ajustement par la mensualité : 1<br>• Ajustement par la durée et la mensualité : 2 |
| **PRT_RGLT** | Prêt réglementé ou aide | Numérique | 1 | CO | Cette rubrique est codifiée de la façon suivante :<br><br>• Crédit réglementé ou aidé bénéficiant d'une aide publique directe ou indirecte, ou crédit au personnel des établissements de crédit : 1<br>• Crédit bénéficiant d'une subvention directe ou indirecte de la part d'une société non financière (par exemple, prise en charge partielle ou totale des intérêts débiteurs) transitant par les comptes de l'établissement financier : 2<br>• Autre cas : 0<br>Il convient de saisir la valeur 0 si le prêt concerné ne bénéficie d'aucune aide ou si son taux n'est régi par aucune réglementation. Il convient également de<br><br>saisir la valeur 0 si le crédit concerné bénéficie d'une aide dont la nature diffère des deux premiers cas. |
| **PRT_RLS** | Prêt relais et travaux | Alpha | 2 | CO | La variable « prêts relais et travaux » permet de qualifier l'objet des contrats de crédits immobiliers, selon la codification suivante :<br><br>Prêt relais – financement de travaux : 00<br><br>Prêt relais – acquisition ancien résidence principale : 01<br><br>Prêt relais – acquisition ancien résidence secondaire : 02 |

| CODE XML | LIBELLE | TYPE | LONGUEUR | PRESENCE (OBligatoire / FAcultarif / COnditionnel) | COMMENTAIRES |
|---|---|---|---|---|---|
| | | | | | Prêt relais – acquisition ancien investissement locatif : 03<br>Prêt relais – acquisition neuf résidence principale : 04<br>Prêt relais – acquisition neuf résidence secondaire : 05<br>Prêt relais – acquisition neuf investissement locatif : 06<br>Prêt classique – financement de travaux : 07<br>Prêt classique – acquisition ancien résidence principale : 08<br>Prêt classique – acquisition ancien résidence secondaire : 09<br>Prêt classique – acquisition ancien investissement locatif : 10<br>Prêt classique – acquisition neuf résidence principale : 11<br>Prêt classique – acquisition neuf résidence secondaire : 12<br>Prêt classique – acquisition neuf investissement locatif : 13 |
| **PRT_RSTR** | Prêt restructuré | Binaire | 1 | CO | La variable « Prêt restructuré » identifie les crédits octroyés dans le cadre d'un rachat de crédit :<br>• Rachat de crédit : 1<br>• Autre objet : 0 |
| **TX_COMM_DEC** | Taux de la commission de découvert | Numérique | 6 | CO | Le taux de la commission de découvert est renseigné sur 6 caractères (4 décimales après la virgule, même s'il s'agit de zéros) et indiqués sans virgule ni point décimal.<br>La valeur du TX_COMM_DEC est positive ou nulle.<br>Précéder le taux de la commission de découvert d'un nombre de 0 suffisant pour que la longueur de la valeur corresponde à la longueur requise. |
| **ZONE_RD** | Zone de résidence | Numérique | 1 | OB | La zone de résidence du client codifiée de la façon suivante :<br>• Bénéficiaire résident : 1<br>• Bénéficiaire non résident mais appartenant à l'un des pays de la zone euro : 0 |
| **MT_REMBRST** | Montant du remboursement | Numérique | 11 | CO | Le montant du remboursement est exprimé en euros, sans décimale.<br>La valeur est strictement positive. |
| **PERIOD_RBRST** | Périodicité de remboursement | Numérique | 1 | CO | La périodicité de remboursement est codifiée de la façon suivante :<br>• Mensuelle : 0<br>• Trimestrielle : 1<br>• Autre : 2 |
| **SURETE** | Type de sûreté | Numérique | 1 | OB | Le type de sûreté garantissant éventuellement le contrat de crédit :<br>• Crédits garantis par des sûretés immobilières : 1<br>• Crédits garantis par des sûretés autres qu'immobilières : 2<br>• Crédits garantis par des sûretés immobilières et autres qu'immobilières : 3<br>• Crédits non garantis : 0 |
| **REVENU_ANN** | Revenu annuel | Numérique | 10 | CO | Le montant du revenu du ménage, sous forme annualisée, utilisé dans le cadre du dossier d'octroi de crédit, est exprimé en euros, sans décimale. |

| CODE XML | LIBELLE | TYPE | LONGUEUR | PRESENCE (OBligatoire / FAcultarif / COnditionnel) | COMMENTAIRES |
|---|---|---|---|---|---|
| | | | | | La valeur est strictement positive. |
| **SIREN** | Numéro SIREN du bénéficiaire | Numérique | 9 | OB | Le numéro SIREN du bénéficiaire doit être un numéro de SIREN valide (Cf. contrôle défini ci-après), 100000009 pour les immatriculations en cours, ou 200000008 pour les bénéficiaires monégasques, ou 999999999 pour les bénéficiaires nonrésidents |
| **MONNAIE** | | Alpha | 3 | | |
| **ACTIVITE** | | Alpha | 3 | | |
| **TYP_POP** | | Alpha | 3 | | |
| **SCT** | Identifiant de la section | Alpha | 4 | OB | Découpage du Tableau correspondant à cinq formulaires identifiés par le code de la population bénéficiaire MCO1 à MCO5 |

*Tableau 8 : Structure du fichier XML de la collecte M_CONTRAN pour le formulaire MCO1*

**Annexe 3 : Analyse de l'intégration du fichier de collecte de l'usure dans ROSTAM**

Pour rappel, les établissements de crédit déposent sur le portail OneGate les fichiers de collecte des données relatives aux calculs de l'usure. Si ces derniers sont conformes au format attendu, ils sont transmis à ROSTAM via EAI pour les intégrer en base de données.

Avant intégration dans ROSTAM, chaque fichier est de nouveau contrôlé. Le fichier sera rejeté par ROSTAM si le fichier n'est pas valide. L'information du rejet est tracée dans ROSTAM et les responsables de collectes sont mis au courant par mail afin de contacter le remettant en question pour corriger l'erreur. Les contrôles des données fonctionnels (exemple : vérifier que le guichet est actif sur l'échéance) ne donne pas lieu à un rejet du fichier si une anomalie est détectée. La ou les anomalies sont tracées en base à titre d'information mais l'ensemble des données sont intégrées. La liste des contrôles effectués est dans le tableau ci-dessous (cf. Tableau 9).

Le chargement de la collecte des données dans ROSTAM est prévu à partir mai 2017 lors de la réception de l'échéance d'avril 2017. Une reprise de données à partir de Juin 2010 est à effectuer. Pour cette reprise, le flux de chargement standard sera utilisé, OneGate pouvant re-générer les fichiers XML associés à ces remises.

Remarque : Avant mon projet, les données relatives à l'usure étaient stockées dans une base de données appelée SISMF (Système d'Information Statistiques Monétaire et Financière).

Les différents contrôles sur le fichier de collecte sont :

| Code | Libelle | Remise | Type | Catégorie | commentaire |
|---|---|---|---|---|---|
| ANO_MCO_ST_01 | La balise <Report n'existe pas | rejetée | Anomalie | Technique | généré par ROSTAM |
| ANO_MCO_ST_02 | La balise <Report n'est pas unique | rejetée | Anomalie | Technique | généré par ROSTAM |
| ANO_MCO_ST_03 | La balise <Data n'existe pas | rejetée | Anomalie | Technique | généré par ROSTAM |
| ANO_MCO_ST_04 | La balise <Item existe alors que balise <Data action = nihil est présente | rejetée | Anomalie | Technique | généré par ROSTAM |
| ANO_MCO_S | La balise <Item n'existe pas | rejet | Anomali | Technique | généré |

| | | | | | |
|---|---|---|---|---|---|
| T_05 | alors que la balise <Data action = nihil n'est pas présente | ée | e | | par ROSTAM |
| CA00010 | Etablissement ou guichet non attendu | chargée | Information | Assujettissement | généré par ROSTAM |
| CA00020 | Guichet manquant | chargée | Information | Assujettissement | généré par ROSTAM |
| CC00001 | Saisie obligatoire de l'identifiant de la section | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00002 | Saisie obligatoire de la référence du crédit | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00003 | Saisie obligatoire de la catégorie de l'instrument financier | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00004 | Saisie obligatoire du montant du crédit | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00005 | Saisie obligatoire de la part dans le pool | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00006 | Saisie obligatoire des conditions de négociation | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00007 | Saisie obligatoire de l'usage du prêt | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00008 | Saisie obligatoire de l'index de référence | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00009 | Saisie obligatoire de la période de fixation initiale du taux (PFIT) | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00010 | Saisie obligatoire du taux effectif au sens étroit (TESE) | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00011 | Saisie obligatoire du taux effectif global (TEG) | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00012 | Saisie obligatoire du champ prêt réglementé ou aide | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00013 | Saisie obligatoire du champ prêt restructuré | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00014 | Saisie obligatoire de la zone de résidence | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00015 | Saisie obligatoire du revenu annuel | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00016 | Saisie obligatoire du numéro SIREN du bénéficiaire | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00017 | Montant maximum autorisé obligatoire si instrument financier = 100, 320 ou 330, interdit sinon | chargée | Information | Collecte MCO | reçu de OneGate |

| | | | | | |
|---|---|---|---|---|---|
| CC00018 | Durée initiale interdite si instrument financier = 100, 320 ou 330, obligatoire sinon | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00019 | Taux d'intérêt maximum obligatoire si IDX_REF différent de 0, interdit sinon | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00020 | Mode d'ajustement obligatoire si IDX_REF différent de 0, interdit sinon | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00021 | Prêt relais et travaux obligatoire si instrument financier = 600, 610, 620, 630, 640, 650, 660, 670, 680 ou 690, interdit si instrument financier = 100, 320 ou 330, facultatif sinon | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00022 | Taux de la commission de découvert obligatoire si instrument financier = 100, interdit sinon | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00023 | Montant du remboursement obligatoire si instrument financier = 300, 310, 440, 500, 510, 600, 610, 620, 630, 640, 650, 660, 670, 680, 690, 700, 800, 900, 910, ou 920, interdit sinon | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00024 | Validation Périodicité de remboursement obligatoire si instrument financier = 300, 310, 440, 500, 510, 600, 610, 620, 630, 640, 650, 660, 670, 680, 690, 700, 800, 900, 910, ou 920, interdite sinon | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00025 | Le montant du crédit doit être une valeur numérique | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00026 | Le montant maximum autorisé doit être une valeur numérique | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00027 | La part dans le pool doit être une valeur numérique | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00028 | La durée initiale doit être une valeur numérique | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00029 | Le taux effectif au sens étroit doit être une valeur numérique | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00030 | Le taux effectif global doit être | charg | Informat | Collecte | reçu de |

| | | | | | |
|---|---|---|---|---|---|
| | une valeur numérique | ée | ion | MCO | OneGate |
| CC00031 | Le CAP doit être une valeur numérique | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00032 | Le taux de la commission de découvert doit être une valeur numérique | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00033 | Le montant du remboursement doit être une valeur numérique | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00034 | Le revenu annuel doit être une valeur numérique | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00035 | Le numéro SIREN du bénéficiaire doit être une valeur numérique | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00036 | Contrôle de la longueur du code guichet | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00037 | Contrôle de la longueur de la référence du crédit | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00038 | Contrôle de la longueur du montant du crédit | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00039 | Contrôle de la longueur du montant maximum autorisé | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00040 | Contrôle de la longueur de la part dans le pool | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00041 | Contrôle de la longueur de la durée initiale | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00042 | Contrôle de la longueur du TESE | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00043 | Contrôle de la longueur du TEG | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00044 | Contrôle de la longueur du CAP | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00045 | Contrôle de la longueur du taux de la commission de découvert | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00046 | Contrôle de la longueur du montant du remboursement | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00047 | Contrôle de la longueur du revenu annuel | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00048 | Contrôle de la longueur du SIREN | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00049 | Contrôle de la catégorie de l'instrument financier (liste) | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00050 | Contrôle des conditions de négociation (liste) | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00051 | Contrôle de l'usage du prêt | charg | Informat | Collecte | reçu de |

| | | | | | |
|---|---|---|---|---|---|
| | (liste) | ée | ion | MCO | OneGate |
| CC00052 | Contrôle de l'index de référence (liste) | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00053 | Contrôle de la PFIT (liste) | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00054 | Contrôle du mode d'ajustement (liste) | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00055 | Contrôle du prêt réglementé (liste) | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00056 | Contrôle du prêt relais et travaux (liste) | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00057 | Contrôle du prêt restructuré (liste) | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00058 | Contrôle de la zone de résidence (liste) | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00059 | Contrôle de la périodicité de remboursement (liste) | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00060 | Contrôle du type de sûreté (liste) | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00061 | Montant du crédit strictement positif | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00062 | Montant maximum autorisé strictement positif | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00063 | Part dans le pool strictement positive et inférieure ou égale à 100 | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00064 | Durée initiale strictement positive | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00065 | Montant du remboursement strictement positif | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00066 | Revenu annuel strictement positif | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00067 | Unicité de la référence du crédit (RFLICR) | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00068 | Contrôle de cohérence entre la PFIT et la durée initiale du crédit | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00069 | Contrôle de cohérence entre le TESE et le TEG | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00070 | Contrôle de cohérence entre le TEG et le taux de l'usure | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00071 | Contrôle de cohérence entre le TESE et le CAP | charg ée | Informat ion | Collecte MCO | reçu de OneGate |
| CC00072 | Contrôle de cohérence entre le montant du revenu annuel (REVENU_ANN) et le montant | charg ée | Informat ion | Collecte MCO | reçu de OneGate |

| | | | | | |
|---|---|---|---|---|---|
| | du remboursement annuel (MT_REMBRST) | | | | |
| CC00073 | Contrôle de cohérence entre le montant du remboursement (MT_REMBRST) et le montant initial emprunté (MT_CRDT), le TEG, la durée initiale ((DUREE_IN) et la périodicité de remboursement (PERIOD_RBRST) | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00074 | Contrôle de validité du numéro SIREN | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00075 | Contrôle de cohérence entre le numéro SIREN (SIREN) et la zone de résidence (ZONE_RD) | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00076 | Saisie obligatoire du type de sûreté | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00077 | Saisie obligatoire du numéro SIREN du bénéficiaire | chargée | Information | Collecte MCO | reçu de OneGate |
| CC00080 | Valeur « nom de la variable dans la RG_MCO_ENR_14 » est erronée. Valeur transformée à vide | chargée | Information | Collecte MCO | reçu de OneGate |
| CC99999 | Code anomalie par défaut | chargée | Information | Collecte MCO | reçu de OneGate |
| INF_MCO_1 | Le CIB XXXXX n'est pas connu du Référentiel ou n'est pas actif pour l'échéance de la remise | chargée | Information | Donnees | généré par ROSTAM |
| INF_MCO_2 | Le guichet XXXXX n'est pas connu du Référentiel ou n'est pas actif pour l'échéance de la remise | chargée | Information | Donnees | généré par ROSTAM |

*Tableau 9 : Liste des contrôles ROSTAM effectués sur le fichier de collecte de l'usure*

**<u>Annexe 4 : Extrait du modèle conceptual de données ROSTAM</u>** (en rouge les indicateurs ajoutées dans la table de fait pour l'usure)

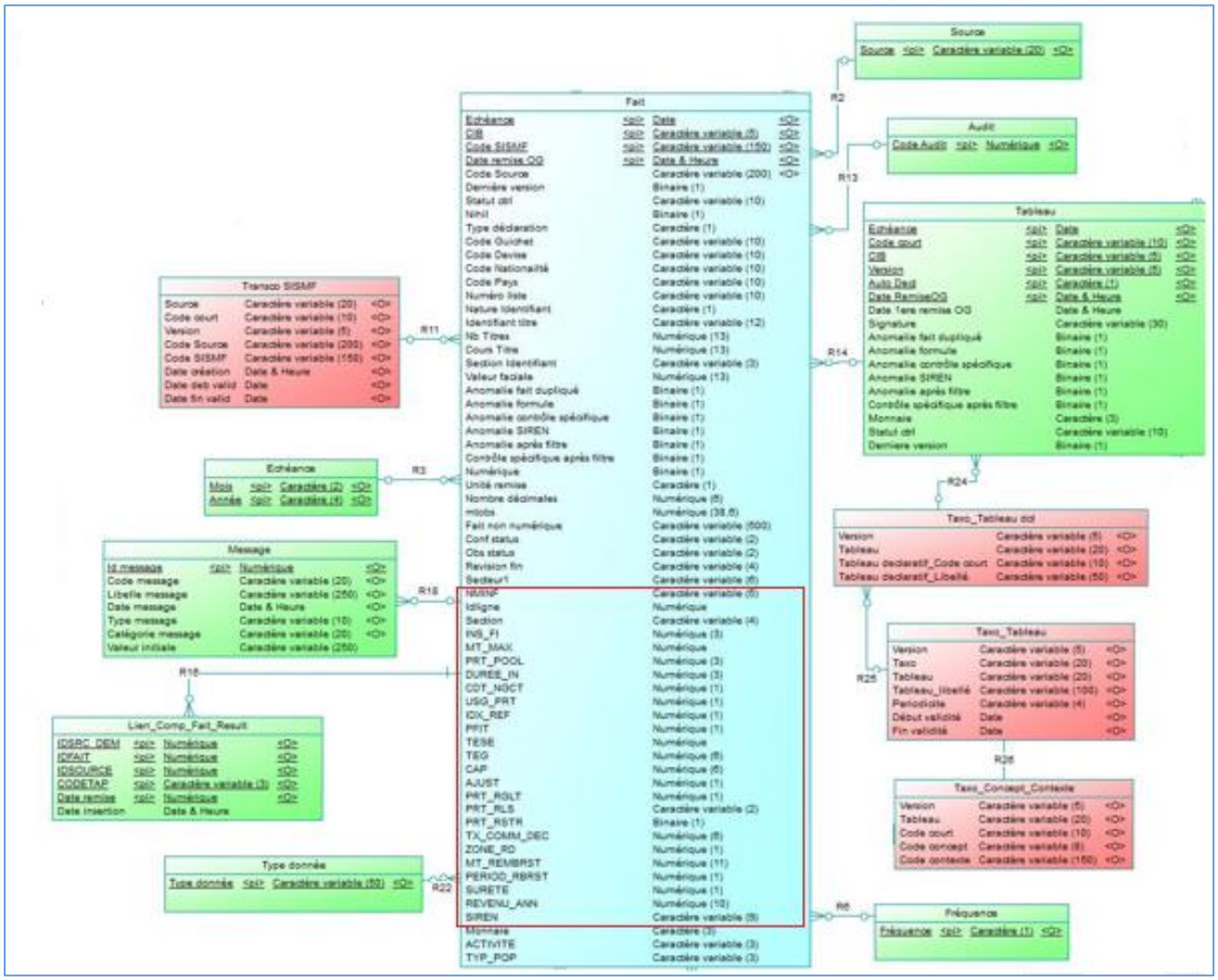


*Figure 89 : Extrait du modèle conceptuel de données ROSTAM*

# RÉSUMÉ

Le mémoire s'articule autour d'une problématique de deux aspects de migration technique :

- Une migration partielle de la chaîne décisionnelle du calcul des taux d'usure ;
- Proposition d'une méthodologie de migration de SAS vers R.

La migration partielle de la chaîne décisionnelle du calcul des taux d'usure implique un nouveau système d'information de stockage des données et une adaptation de la chaîne de calcul SAS. En tant que chef de projet responsable des évolutions SAS, j'ai été en charge d'analyser les impacts, d'estimer la charge nécessaire, de l'évolution des spécifications techniques, de l'évolution des programmes SAS et de la recette.

La Banque de France s'oriente vers une diminution de l'utilisation du logiciel SAS et une augmentation de celle du logiciel R. Dans l'objectif d'obtenir une estimation du temps de migration de SAS vers R, j'ai réalisé un outil d'estimation de charge avec le langage Shell et SAS. Il analyse les programmes aux travers de divers indicateurs. Il en résulte un score de complexité, une estimation de charge en nombre de jours homme et une classe de complexité de migration.

# MOTS CLÉS

EAI, SAS, R, taux d'usure, Informatique Décisionnelle, migration, modèle en étoile

# ABSTRACT

The project deals with two aspects of technical migration :

- Partial migration of the decision-making chain in the calculation of wear rates ;
- Proposal of a migration methodology from SAS to R.

The partial migration of the decision-making chain of wear rate calculation involves a new information system and an adaptation of the SAS calculation chain. As a project manager responsible for SAS evolutions, I was in charge of analyzing the impacts, estimating the workload, the evolution of the technical specifications, the evolution of the SAS programs and the tests.

The Banque de France is moving towards a decrease in the use of SAS, unlike R which is recording a substantial increase. In order to obtain an estimation of the migration time from SAS to R, I conceived a software that estimates the workload with Shell and SAS Language. It analyzes the programs through various indicators. The result is a complexity score, a workload and a migration complexity class.

# KEY WORDS

EAI, SAS, R, wear rates, Business Intelligence, migration, star model