



GLOSSAIRE

MT	Maroc Telecom
API	Application Programming Interface
BD	Base de Données
BSCS	Business Support & Control System
NMT	Nordic Mobile Telephone
GSM	Global System for Mobile
DCS	Digital Communication System
Neto	Netonomy
OVW	OVERWRIT
NA	Nouvel Abonnement
SQL	Structural Query Langage
XML	eXtensible Markup Language
ANSI	L'American National Standards Institute
UML	Unified Modeling Language





LISTE DES FIGURES

Figure 1: Implantation d'Atos Origin dans le monde.....	11
Figure 2: Historique d'Atos Origin.....	12
Figure 3: Organigramme d'Atos Origin.....	13
Figure 4: Domaine d'expertise d'Atos Origin.....	15
Figure 5 : Modules Batch.....	16
Figure 6 : Producteur –consommateur.....	18
Figure 7: Architecture de Spring Batch.....	29
Figure 8: Cardinalités entre Job, JobInstance et JobExecution.....	31
Figure 9: Diagramme de classes client/contrat.....	36
Figure 10: Diagramme de classes contrat/service.....	37
Figure 11: Diagramme de classes factures.....	37
Figure 12: Diagramme de classe promotions.....	38
Figure 13: Résultat obtenu avec Eclipse.....	59
Figure 14: Résultat obtenu avec Oracle de la méthode. GetSncode(2).....	60
Figure 15: Résultat obtenu avec Oracle de la méthode GetCatCustomer(5).....	60
Figure 16: Résultat obtenu avec Oracle de la méthode GetContractInfo(6).....	60
Figure 17: Résultat obtenu avec Oracle de la méthode GetLastContractRp(7).....	61
Figure 18: Résultat obtenu avec Oracle de la méthode GetOverWriteInfo(17).....	61
Figure 19: Résultat obtenu avec Oracle de la méthode GetSrvParam(8).....	61
Figure 20: Résultat obtenu avec Oracle de la méthode GetLastSrvSatus (11).....	61
Figure 21: Résultat obtenu avec Oracle de la méthode CheckPromoAssigned (14).....	62
Figure 22: Résultat obtenu avec Oracle de la méthode SetOverWriteSrv(16) avant l'application de l'OVW.....	62
Figure 23 : Résultat obtenu avec Oracle de la méthode SetOverWriteSrv(16) après l'application de l'OVW.....	62

LISTE DES TABLEAUX

Tableau 1: Attributs de la classe CUSTOMER_ALL.....	39
Tableau 2: Attributs de la classe PRICEGROUP_ALL.....	39
Tableau 3: Attributs de la classe BICYCLES.....	39
Tableau 4: Attributs de la classe REASONSTATUS_ALL.....	40



Tableau 5: Attributs de la classe RATEPLAN.....	40
Tableau 6: Attributs de la classe RATEPLAN_HIST	40
Tableau 7: Attributs de la classe FEES.....	41
Tableau 8: Attributs de la classe CONTRACT_ALL	41
Tableau 1: Attributs de la classe CONTRACT_HISTORY.....	42
Tableau 10: Attributs de la classe CONTRACT_SERVICE	42
Tableau 11: Attributs de la classe MPUSNTAB	42
Tableau 12: Attributs de la classe PROFILE_SERVICE	43
Tableau 13: Attributs de la classe PR_SERV_STATUS_HIST.....	43
Tableau 14: Attributs de la classe PR_SERV_SPCODE_HIST.....	44
Tableau 15: Attributs de la classe PARAMETER_VALUE	44
Tableau 16: Attributs de la classe ORDERHDR_ALL	44
Tableau 17: Attributs de la classe ORDERTRAILER	45
Tableau 18: Attributs de la classe MPULKTMB	45
Tableau 19: Attributs de la classe RULE_PACK.....	46
Tableau 20: Attributs de la classe PROMO_PACK.....	46
Tableau 21: Attributs de la classe PROMO_ASSIGN.....	47
Tableau 22: Attributs de la classe PROMO_ASSIGN_STATE	47
Tableau 23: Attributs de la classe PROMO_MECH_SET	47
Tableau 24: Attributs de la classe PROMO_MECH_SET_DATA.....	48
Tableau 25: Attributs de la classe PROMO_QUAL_RESTRICT	48

SOMMAIRE

Dédicaces.....	1
Remerciements.....	2
Glossaire.....	3
LISTE DES FIGURES.....	4
LISTE DES TABLEAUX.....	5
SOMMAIRE.....	6
Introduction	8
Chapitre 1: Cadre général du projet.....	10





I. Présentation générale d'ATOS ORIGIN.....	11
1. Présentation.....	11
2. Historique.....	11
3. Organigramme.....	12
4. Activité.....	13
5. Domaines d'expertises.....	15
II. Système d'information et de facturation de Maroc Telecom.....	15
1. Architecture du système BSCS.....	15
2. Modules BSCS.....	16
3. DaTA.....	17
4. Producteur – Consommateur.....	18
Chapitre 2 : Analyse de l'existant et environnement technique.....	20
I. Promotions de MT.....	21
1. Promo 50% fixe Mars 2011.....	21
2. Promo Data et Internet-Mars2011.....	24
II. But d'automatisation des traitements.....	26
1. problématique.....	26
2. Solution demandée	26
III. Environnement Technique.....	27
A. Framework utilisé Spring / Spring Batch.....	27
1. Présentation de Spring.....	27
2. Présentation de Spring Batch.....	28
3. Pourquoi Spring Batch.....	29
4. Architecture de Spring Batch.....	29
5. Les principaux éléments de Spring batch.....	30
B. Environnement de développement.....	32
1. Eclise.....	32
2. Toad For Oracle.....	32
C. Langages utilisés.....	32
1. Java.....	32
2. SQL/PL SQL.....	32



Chapitre 3 : Etude de la BD de MT	34
I. Analyse et conception.....	35
1. Diagrammes de classes.....	36
2. Caractéristiques des tables.....	39
Chapitre 4 : Mise en œuvre	49
I. Description du projet.....	50
1. Description des méthodes développées.....	50
2. Atelier.....	56
3. Organigramme de la classe qui fait appel aux requêtes.....	58
II. Vérification et test.....	59
Conclusion	63
Perspective.....	64
Bibliographie et Webgraphie.....	65





Introduction

Le Maroc, à l'instar des pays occidentaux, connaît un véritable développement dans le domaine de l'informatique, qui semble être de nos jours primordial dans le monde de l'entreprise.

Ainsi l'informatique qui a pour but de tout automatiser est devenue un outil vital pour l'entreprise, et l'informatisation peut conduire à d'importants gains de productivité, à une amélioration de la qualité et aussi à une réduction des coûts. C'est dans cette perspective qu'Atos Origin évolue et propose à ses clients au travers de ses services et prestations.

De ce fait, l'opérateur marocain de télécommunications, Maroc Telecom(MT), leader sur l'ensemble de ses segments d'activités (téléphonie fixe, mobile et internet) a confié à son partenaire idéal Atos Origin, en raison de ses compétences et de son expérience dans de précédents projets, un projet qui *va apporter des bénéfices en termes de coût, délai et qualité d'activation, afin de maintenir un niveau élevé de satisfaction de ses clients.*

Cet opérateur propose souvent des promotions fixes afin de fidéliser ces clients et attirer de nouveaux abonnés. Cependant ces promotions qui sont développées par Atos Origin de tel sorte que la promotion 1 est développée par un développeur 1, la promotion 2 est développée par un développeur 2, la promotion 3 est développée par un autre développeur 3 et ainsi de suite, courent les mêmes étapes de développement et subissent un traitement répétitif, ce qui engendre une perte au niveau du coût et du temps de développement, ainsi qu'une grande charge au niveau du travail à effectuer.

Pour éviter ces traitements répétitifs, pourquoi ne pas créer une API qui factorise, automatise et industrialise l'ensemble de ces promotions fixes de MT?





C'est dans ce sens que s'inscrit mon projet de fin d'études intitulé « l'automatisation et l'industrialisation des promotions fixe de Maroc Telecom » visant à mettre en œuvre cette solution.

Ce projet porte deux grands volets :

- La factorisation : qui consiste à créer un fichier jar qui contient toutes les méthodes répétitives effectuées pour chaque promotion.
- L'industrialisation : qui consiste à adapter l'application selon le besoin du développeur.

Ma mission était sous forme de: d'abord, d'analyser des exemples de promotions offertes déjà par MT, comparer les tâches de chaque promotion afin de déduire le traitement récidivant, puis d'étudier la base de données (BD) de MT, se focaliser sur les 25 tables nécessaires pour appliquer les promotions fixes et inférer la relation entre ces différentes tables. Par la suite effectuer la réalisation de cette application.

Le présent rapport trace les phases du déroulement du projet. Il est organisé en quatre chapitres.

- Le premier définit le contexte général du projet. Il est composé de deux parties: la première présente l'organisme d'accueil (ATOS ORIGIN), tandis que la seconde traite l'organisation de l'architecture du système d'information et de facturation de MT.
- Le deuxième chapitre porte sur l'analyse des exemples de promotions fixes, la présentation de la problématique et la solution à adopter ainsi que l'environnement technique utilisé.
- Le troisième chapitre du rapport est consacré à l'étude de la BD de MT.
- Le quatrième et dernier chapitre, porte sur la réalisation de l'application. Il est composé de deux parties: la première consiste à décrire les différentes méthodes à implémenter dans l'API ainsi que l'atelier utilisé. Quant à la deuxième, elle représente quelques prises d'écrans de cette application comme démonstration.

Et je termine par une conclusion qui synthétise le travail réalisé.





Chapitre 1 :

Cadre général du projet

Ce chapitre présente le contexte général du projet. Il présente d'abord L'organisme d'accueil « Atos Origin » ensuite il décrit le système de facturation de MT.

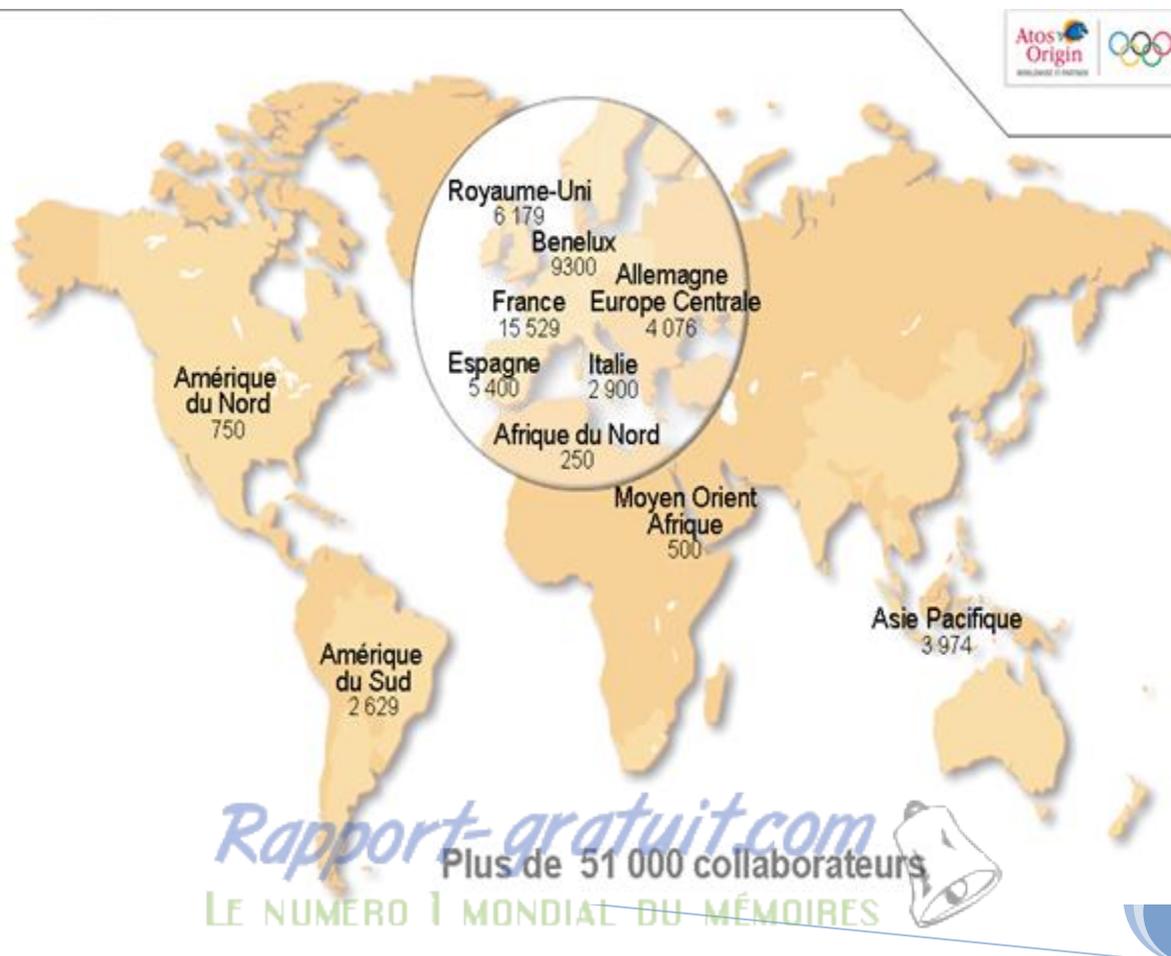


III. Présentation générale d'Atos Origin [1]

6. Présentation

Atos Origin est l'un des principaux acteurs internationaux du secteur des services informatiques. Sa mission est de traduire la vision stratégique de ses clients en résultats par une meilleure utilisation de solutions de conseil, intégration de systèmes et infogérance.

Implanté dans 40 pays à travers le monde, Atos Origin réalise un chiffre d'affaires annuel de 5,8 milliards d'euros avec un effectif de plus de 51 000 personnes. Atos Origin est le partenaire informatique mondial des Jeux Olympiques. Il exerce ses activités sous les noms d'Atos Origin, Atos Worldline et Atos Consulting.



2. Historique

Atos résultait du rapprochement intervenu en 1997 entre deux sociétés françaises de services informatiques, Axime et Sligos. Au moment de sa fusion avec Origin, Atos comptait un effectif de 11 000 salariés et générait un chiffre d'affaires annuel de 1,1 milliard d'euros environ.

Origin était une filiale de Royal Philips Electronics, qui avait été créée en 1996 par le biais d'une fusion entre BSO/Origin et Philips Communications. Au moment de sa fusion avec Atos en octobre 2000, Origin comptait un effectif de 16 000 salariés répartis dans 30 pays dans le monde et générait un chiffre d'affaires annuel de 1,6 milliard d'euros environ.

En août 2002, les activités de KPMG Consulting au Royaume-Uni et aux Pays-Bas ont été rachetées et regroupées. Cette opération a conféré au Groupe une présence majeure sur le segment du conseil au sein du marché des services informatiques.

En janvier 2004, Atos Origin a acquis le Groupe Sema auprès de Schlumberger, donnant ainsi naissance à l'un des leaders internationaux des services informatiques. Au moment de l'acquisition, le Groupe Sema employait 20 000 personnes et enregistrait un chiffre d'affaires annuel de quelque 2,4 milliards d'euros. L'effectif d'Atos Origin s'élevait alors à 26 500 salariés, pour un chiffre d'affaires annuel supérieur à 3 milliards d'euros.

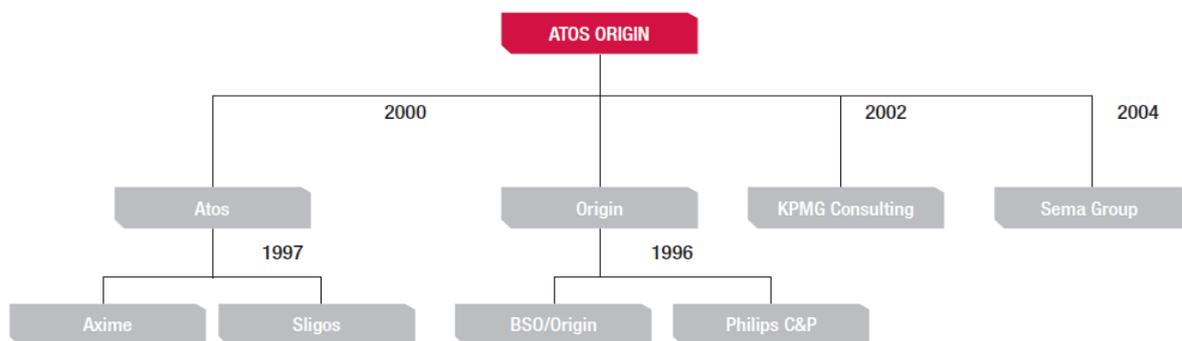
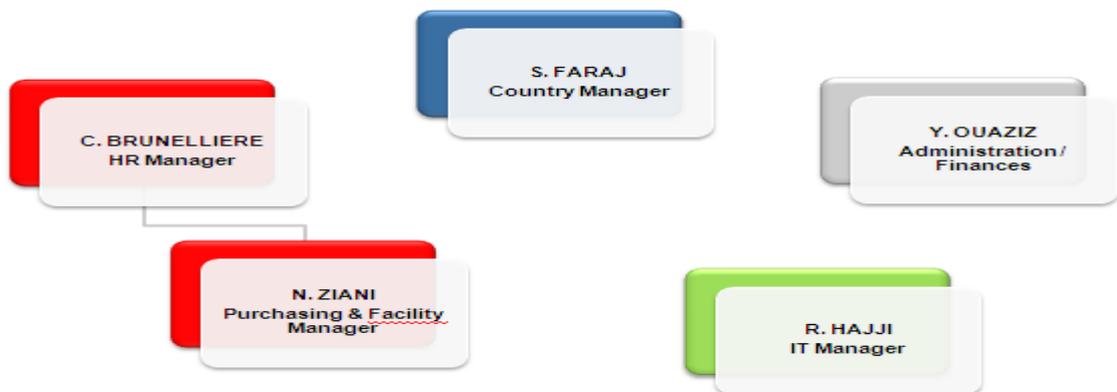


Figure 2 : Historique d'Atos Origin

3. Organigramme

L'organigramme d'Atos Origin se présente comme suit :





L'équipe technique :

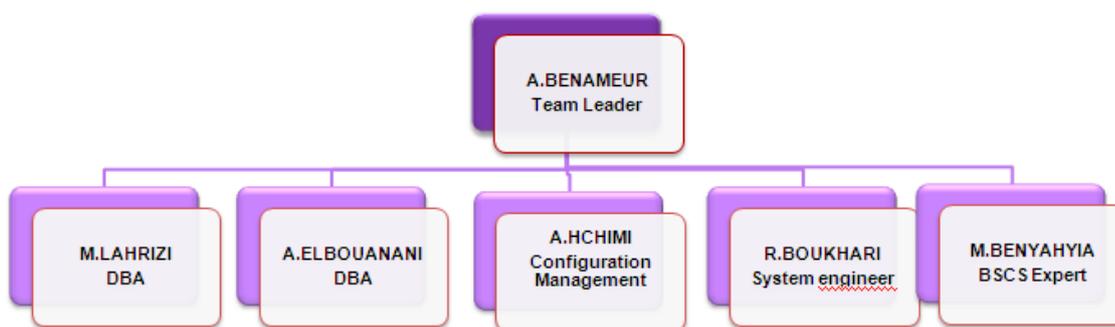


Figure 3: Organigramme d'Atos Origin

Dans le but d'accélérer sa croissance et de mondialiser son fonctionnement. Atos Origin a annoncé l'extension de ses activités au Maroc avec la création deux implantations:

- Le siège à Rabat depuis 2003.





- Centre de services nearshore à Casablanca destiné à ses clients francophones depuis 2007.

4. Activités

Atos Origin se base sur quatre activités :

a. Conseil:

Le Groupe offre des services et solutions, allant du développement de stratégies jusqu'au choix de solutions et de technologies appropriées.

Ses clients sont ainsi en mesure d'améliorer leur productivité et de générer d'avantage de valeur grâce à une approche innovante des processus métiers, conjuguée à une intégration performante des technologies, ainsi que par des investissements stratégiques en ressources humaines. Atos Consulting s'assure que tous les aspects d'une organisation « ressources humaines, processus et technologie » sont entièrement alignés sur la stratégie de l'entreprise.

b. Intégration des systèmes:

Atos Origin a une vaste expérience en matière d'intégration des personnes, des processus et des technologies, ce qui lui permet de concevoir, de bâtir et d'exploiter des solutions pratiques et robustes. Ainsi, ses spécialistes travaillent avec les clients sur le développement, Le déploiement et la maintenance de leurs systèmes. L'intégration des systèmes ne consiste pas uniquement à intégrer de nouvelles solutions, mais également à tirer le meilleur parti des applications en vue de rentabiliser l'investissement informatique existant.

c. Infogérance:

Atos Origin prend en charge la gestion des infrastructures informatiques clés de ses clients : centres de données, assistance micro-informatique, parcs de serveurs et réseaux de communication.

La société fournit, à travers son réseau mondial, des services accessibles 24h/24 et 7j/7, elle dispose d'une expérience inégalée en matière de déploiement de solutions complexes. Pour assurer un service de premier ordre, Atos Origin s'appuie sur sa propre méthodologie de prestation de service continu tout au long du processus d'évaluation, de planification, de mise en œuvre et de transition.



d. Service transactionnels à haute valeur ajoutée:

Au travers de sa filiale Atos Worldline, Atos Origin est le leader européen dans les services de paiements et les transactions électroniques. Atos Worldline fournit des solutions et services innovants qui permettent d'améliorer la performance de ses clients. Atos Worldline offre également une forte capacité de traitement industriel permettant de traiter des milliards de transactions électroniques.

5. Domaines d'expertises

Atos Origin a une vraie expérience dans les systèmes d'informations, et elle est présente dans plusieurs domaines :



Figure 4: Domaine d'expertise d'Atos Origin

IV. Système d'information et de facturation de MT[2]

Dans le but d'offrir des services de haute qualité, MT offre une solution qui permet de fournir à ses abonnés un service d'information et de facturation flexible tout en améliorant le



recouvrement et en limitant les pertes. Ce service est basé sur un système appelé BSCS: Business Support & Control System.

1. Architecture du système BSCS

L'architecture de BSCS se base sur le modèle client-serveur.

Serveur:

- OS: UNIX.
- Code: C/C++.
- BD Oracle.
- Modules batch: valorisation et facturation.

Client:

- OS: Windows.

Il existe deux architectures différentes de BSCS à MT: mobile et fixe.

a. Mobile

Ce type d'architecture est adopté par: Nordic Mobile Telephone abrégé en NMT, GSM 900 (Global System for Mobile), DCS 1800, carte prépayée.

L'architecture mobile a pour application principale :

- GE: la gestion des encaissements.
- SR: la gestion du service de réclamation.
- FIDELIO : service a points.
- IVR : serveur vocal interactif.
- IOG : réalise les activations en masse.

b. Fixe

Ce type d'architecture est adopté par : ISDN ou RNIS, VPN, PBX. Il a pour fonction la gestion des clients gérés par Netonomy(Neto).

2. Modules BSCS

a. Vue générale :



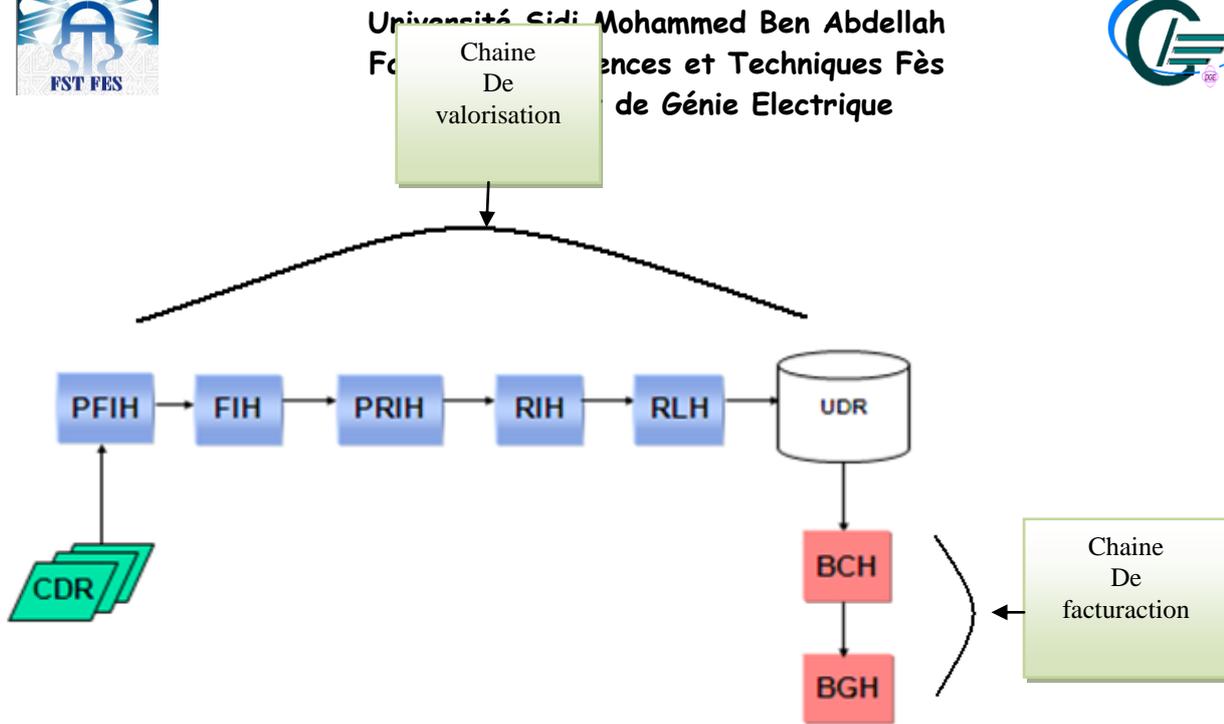


Figure 5: Modules Batch

b. Modules Batch

PFIH (Pré- File Input Handler):

- Reçoit les tickets CDR collectés par CADOT.
- Enregistre les tickets traités dans la table THUFITAB.

FIH (File Input Handler):

- Conversion du format CDR au format UDR.

PRIH (Pré- Rate Input Handler):

- Analyse chaque communication figurant dans l'UDR pour déterminer à quel type de scénario il correspond.

RIH (Rate Input Handler):

- Analyse service, destination, jour, tranches horaires.
- Calcul des coûts des communications.

RLH (Rate Load Handler):

- Charge en BD les UDRs des communications valorisées.

BCH (Bill Generation Handler):

- Calcul des montants des factures.
- Génération des messages TIMM EDIFACT.

BGH (Bill generation handler):

- Lecture des messages TIMM.





- Génération de la facture.

3. DaTA (Data Transmission Application)

La valorisation des communications est réalisée par des modules Batch, qui échangent des informations à travers un bus de communication: DaTA.

Un fichier d'appel (CDR) qui arrive du réseau est traité par plusieurs batch, dans l'ordre de traitement : PFIH, FIH, PRIH, RIH, RLH, chacun d'eux effectue une opération visant à modifier le format du ticket arrivant, ou enrichir son contenu. Le résultat final est ensuite stocké dans une BD pour la facturation.

a. Rôles du DaTA

Le DaTa est responsable de la distribution des informations qui passent par la chaîne de valorisation. Il gère des files de fichiers d'informations à traiter.

Ce DaTa a pour rôle:

- L'administration des applications productrices-consommatrices.
- La transmission des données.
- La gestion des reprises des applications productrices-consommatrices.
- La gestion de la persistance des données.

b. Mécanisme de Reprise de DaTA

En cas d'arrêt brutal d'une application productrice-consommatrice, le DaTA gère la persistance des données de la transaction (non-commitée), il relance l'application productrice-consommatrice et reprend les échanges des données de la transaction entre DaTA et l'application productrice-consommatrice.

4. Producteur-Consommateur

Le système producteur-consommateur avec un médiateur a pour principe d'éviter la famine ou l'inter-blocage.



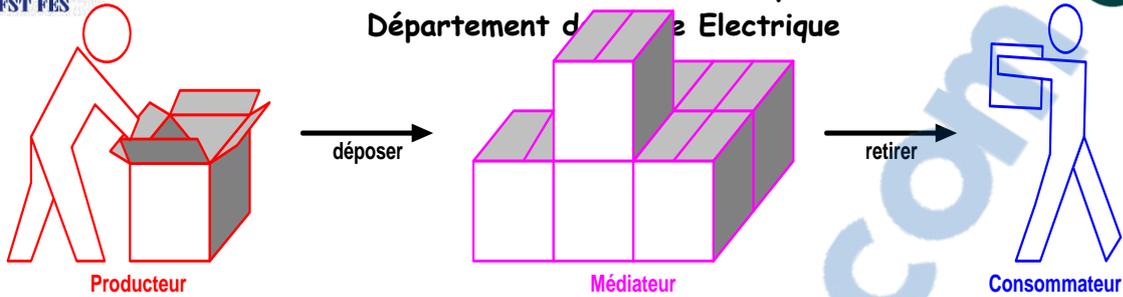


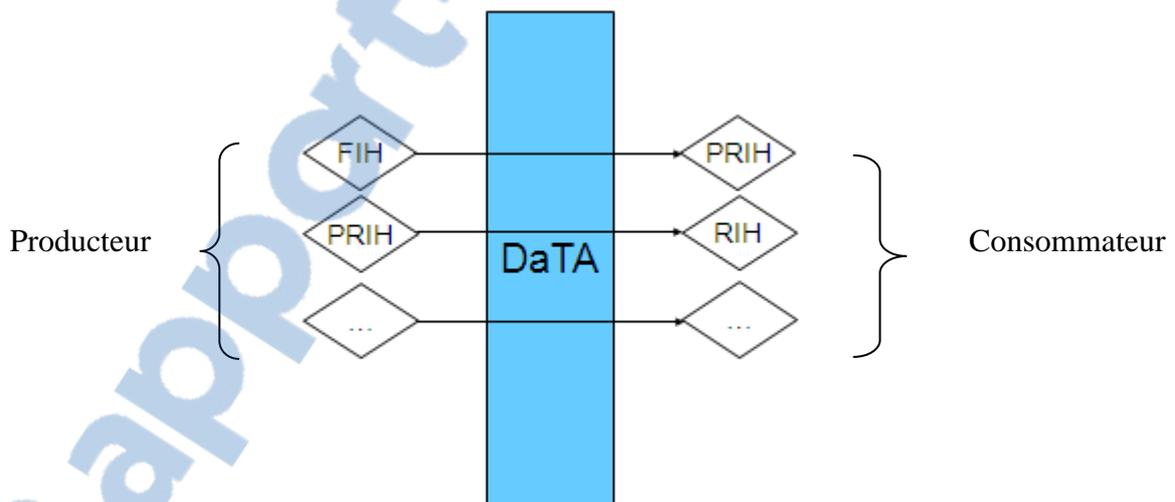
Figure 6: producteur –consommateur

Producteur: Module offrant une information qu'il est capable de générer. Tous les modules batch de BSCS (PFIH, FIH, PRIH, ...) jouent le rôle d'un producteur.

Consommateur: Module ayant besoin d'une information. Tous les modules batch de BSCS (PFIH, FIH, PRIH, ...) jouent le rôle d'un producteur.

Médiateur: Mécanisme permettant :

- L'inscription des producteurs et consommateurs avec les informations susceptibles d'être échangées.
- Mécanisme de gestion de persistance de données.



Conclusion :

Ce chapitre m'a permis d'avoir une idée générale sur l'organisme d'accueil Atos Origin et ses différentes activités ainsi qu'une idée sur le système d'information et de facturation de l'opérateur MT qui fera l'objet d'une automatisation dans ce projet.





Chapitre 2 :

*Analyse de l'existant et
environnement technique*

Ce chapitre présente des exemples de promotions fixes récentes de MT, la problématique ainsi que l'environnement technique utilisé pour le développement de l'application.





I. Promotion de MT

L'évolution et la croissance du secteur de télécommunications, téléphonie et internet, a relevé le défi de la concurrence. Cette concurrence, qui est au bonheur du client, devient plus rude et les marges entre les différents opérateurs se resserrent. Dans ce cadre et pour faire face à cette concurrence, et maintenir sa croissance et son leadership, MT améliore le rapport qualité prix des produits et des services offerts aux clients. Ainsi MT adopte une baisse des prix, une diversification de l'offre, des spots publicitaires, des actions de sponsoring, et surtout des actions de promotions qui visent à fidéliser et satisfaire les clients, recruter de nouveaux abonnés et renforcer sa position d'opérateur innovant.

Parmi les promotions récentes offertes par MT, on trouve promo 50% FIXE MARS 2011, promo Data et Internet-Mars2011 et promo duo phony et ADSL. Dans le paragraphe qui suit, je présenterai les deux premiers cas à titre d'exemples.

1. Promo 50% fixe Mars 2011 [3]

a. Contexte

Lancée en Mars 2011, cette offre s'est appliquée aux abonnés ayant nouvellement souscrits à un abonnement voix (RTC/Marnis) en engagement 12 mois ou 24 mois entre la période du 1er mars au 30 avril 2011. Ainsi ces abonnés ont eu une réduction de 50% sur le montant des 3 premières factures de la nouvelle ligne fixe souscrite.

Cette offre a concerné les catégories résidentielles, professionnelles (cybers, annonceurs, revendeurs agréés, revendeurs non agréés...), entreprises (offices et établissements publics, ambassades, banques.....) et VIP. Dans ce cadre Atos a proposé une solution pour répondre au besoin de MT qui est formulée sous le référentiel FC4240.

b. Proposition Atos

La promotion a été attribuée par un batch, cette évolution a eu un impact sur BSCS. Les contrats éligibles concernés par cette promotion ont été extraits de Neto et chargés dans une table de traitement.





Atos a été mené à créer un nouveau package promotionnel respectant les règles suivantes:

- ❖ Applicable à tout contrat ayant souscrit à une nouvelle ligne RTC et Marnis.
- ❖ Applicable sur les trois premières factures.

c. Diagramme de flux :

Les tâches élémentaires du batch sont les suivantes :

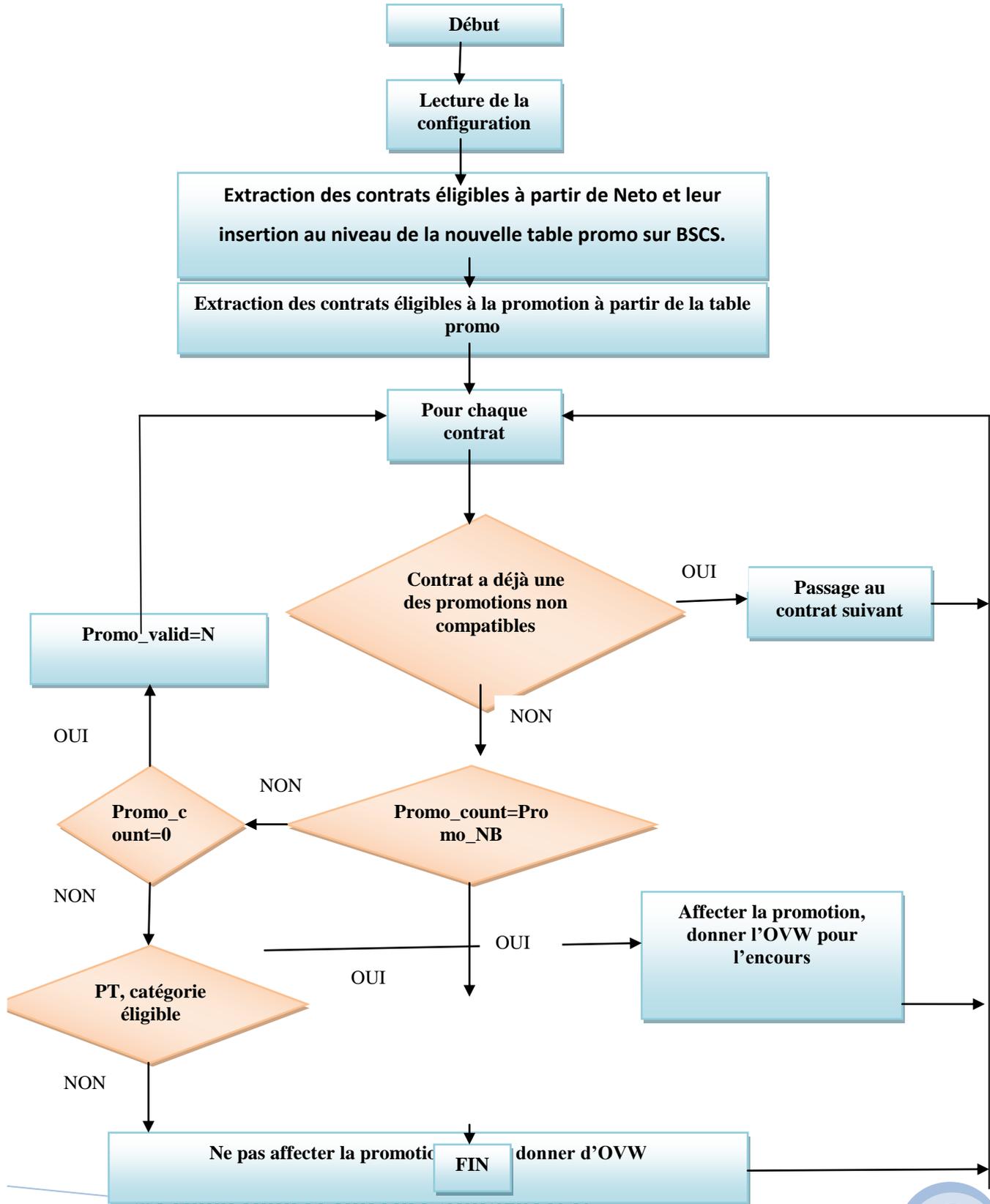
➤ Lecture de la configuration

- RUNDATE: date de lancement du batch (par exemple 01/03/2011).
- DEBUT_PROMO: date de début de la promotion.
- FIN_PROMO : date de fin de la promotion.
- CATEGORIES: catégorie du client.
- ENGAGEMENT : 12, 24.
- PT : plan tarifaire.
- REDUCTION : réduction à appliquer en pourcentage.
- DUREE: durée de la promotion en mois.
- SN: liste des services exclus de l'application de la promotion (SHDES1, SHDES2,...).
- PROMO : liste des promotions non-compatibles (PROMOID1, PROMOID2...).

➤ Traitement

- Extraction des contrats éligibles à partir de Neto et leur insertion au niveau de la nouvelle table promo sur BSCS.
- Extraction des contrats éligibles à partir de la table de la promotion.
- Tests de contrôle pour chaque contrat :
 - ➔ Vérification d'existence des promotions non compatibles.
 - ➔ Vérification du nombre de fois restant pour l'attribution de la promotion pour la même période. Si ce nombre est strictement inférieur à 3 alors on passe à la vérification suivante. Sinon, on arrête le traitement pour le contrat concerné.
- Si le contrat est éligible à la promotion (catégorie et plan tarifaire (PT)).
 - ➔ Assignation de la promotion pour la période facturée.
 - ➔ Attribution de la remise sur l'avance via OVERWRIT (OVW) pour le mois en avance.

Le diagramme suivant explique plus en détail la logique du batch.



a. Contexte



Cette promotion a consisté à faire une relance de la promotion « Promo DATA et Internet » permettant aux :

- Anciens clients data et internet ayant fait des augmentations de débit, de payer le tarif du nouveau débit au prix de l'ancien débit pendant 3 mois.
- Clients qui ont fait de nouvelles souscriptions aux produits data et internet, une remise de 50% sur l'abonnement pendant les trois premiers mois. Cette offre a concerné les catégories entreprises et professionnelles.

Dans ce cadre Atos a proposé une solution pour répondre au besoin de MT qui est formulée sous le référentiel FC4238.

b. Proposition Atos

Il s'agissait d'une relance de cette promotion qui était déjà industrialisée (FC3984) comme suit :

- Un batch coté Neto, qui prend en entrée un fichier de configuration qui contient les PT et les services non concernés, la période de la promotion.... A la base de ce fichier de configuration, il extrait toutes les commandes créés durant cette période. Ces commandes sont enregistrées dans une table de communication.
- Un script côté BSCS qui prend en entrée un fichier de configuration qui contient la date d'insertion de l'OCC. Ce script est appliqué sur les commandes de la table de communication.

Spécificité de cette offre par rapport à celle de la FC3984 :

Les frais d'augmentation de débit et les frais d'accès n'ont pas été annulés et de nouveaux PT ont été aussi impactés.

La solution a consisté donc à modifier les livrables de la FC3984 de tel sorte à prendre en charge la non annulation des frais d'augmentation sur la promotion augmentation de débit, et la non annulation des frais d'accès sur la promotion nouvel abonnement(NA). Cette évolution a eu un impact sur BSCS.

c. Diagramme de flux





Université Sidi Mohammed Ben Abdellah
Faculté Des Sciences et Techniques Fès
Département de Génie Electrique



Les tâches élémentaires du batch sont les suivantes :

➤ **Lecture de la configuration** :

- RUNDATE : date de lancement du batch (par exemple 01/10/2010).
- DEBUT_PROMO: date de début de la promotion.
- FIN_PROMO : date de fin de la promotion.
- Catégories des clients éligibles à la promo NA.
- La liste des PT éligibles à la promotion.
- DUREE: durée de la promotion en mois.
- SN: Liste des services exclus de l'application de la promotion (SHDES1, SHDES2,...).

➤ **Traitement**

- Extraction des clients éligibles à du Neto, et leur insertion au niveau de la nouvelle table promo sur BSCS.
- Extraction des clients éligibles à partir de la table promo sur BSCS.
- Tests de contrôle pour chaque client selon les critères suivants :
 - ➔ Vérification de la mise en service de la commande sous Neto et la récupération du co_id.
 - Si la commande n'est pas mise en service on passe à la commande suivante.
 - Sinon, on récupère le co_id et on passe aux autres vérifications.
 - ➔ Vérification du nombre de fois le client a bénéficié de la promotion pour la même période. Si ce nombre est strictement inférieur à trois alors on passe à la vérification suivante. Sinon, on arrête le traitement pour le client concerné.
- Si le client est éligible à la promotion (catégorie et PT).
 - ➔ Assignation de la promotion non standard pour la période facturée.
 - ➔ Attribuer l'OCC.

La fréquence de lancement de ce batch est mensuel.





II. But de la factorisation des traitements :

À travers ces deux exemples on remarque que pour développer le batch de chaque promotion, Atos Origin suit la même démarche que se soit pour une promotion appliquée sur un contrat, ou promotion appliquée sur un client.

1. Problématique.

Pour réaliser les taches du batch de chaque promotion, Atos Origin effectue un traitement répétitif similaire au traitement des promotions précédentes par ex:

- Chercher la catégorie d'un client.
- Chercher l'état d'un contrat (actif, suspendu, résilié...).
- Chercher le PT d'un contrat (Ex : PT soir & weekend, internet ADSL 1mois 100dh, ADSL pro).
- Ajouter un service pour un contrat.
- Désactiver un service pour un contrat.

Dans ce sens Atos vise à créer une API qui englobe le traitement récurrent des promotions.

2.Solution demandée

C'est dans ce cadre qu'intervient mon projet de fin d'études, qui a pour objectif la création de cette API qui doit centraliser le traitement des promotions, de façon à ce que le développeur qui va développer la promotion n'a rien qu'importer le fichier jar de l'application et adapter l'API selon son besoin.

Cette application a pour but d'éviter de réécrire des centaines de fois le même code dans chaque promotion, afin de réduire le temps de développement, diminuer le volume de travail à réaliser, laminer les sources d'erreurs et la complexité des applications, et améliorer leurs performances, et donc faciliter le travail du programmeur en lui fournissant l'outil de base nécessaire à son travail tout en assurant une grande fiabilité.

L'API qui va être créée doit aborder d'une part le volet technique que doit respecter toutes les routines et les fonctions à développer, et d'autre part les fonctionnalités à implémenter tout en respectant les exigences techniques requises.





a. Exigences techniques de l'API

Cette API doit permettre de :

- Tracer les traitements répétitifs et les traitements nécessitant des calculs (prorata).
- Distinguer les traitements fonctionnels.

b. FONCTIONNALITE UTILE A IMPLEMENTER

Parmi les besoins fonctionnels qu'on peut factoriser sont:

❖ Insérer OCC:

INPUT: co_id, montant, periodicité, date d'effet, sncode.

OUTPUT: customer_id.

❖ OVW:

INPUT: montant de l'OVW, co_id, sncode (sncode ou bien shdes du service), type d'OVW (absolue ou relatif).

OUTPUT: Code retour (OK ou KO).

❖ AFFECTER PROMO:

Assigner promo à un client.

Assigner la promotion à un client. Il faut vérifier si le client a déjà la promotion ou non.

III. Environnement Technique

A) Framework utilisé Spring / Spring Batch [5]:

Afin de réaliser un projet solide, structuré et facilement évolutif, le besoin d'un Framework solide devient indispensable, d'où la nécessité de travailler avec le Framework open source Spring Batch.

1. Présentation de Spring

Spring est un Framework open source pour les applications 3-tiers, facilitant leurs développements et les tests.

Il prend en charge :

- la création d'objets.





- leur mise en relation par l'intermédiaire d'un fichier de configuration qui décrit les objets à fabriquer.
- les relations de dépendances entre ces objets.

Il s'appuie principalement sur l'intégration de trois concepts clés :

- a) l'inversion de contrôle: est assurée de deux façons différentes: la recherche de dépendance et l'injection de dépendance.
 - La recherche de dépendance : consiste pour un objet de trouver ses dépendances avec les autres objets.
 - L'injection de dépendances : l'idée de l'injection des dépendances est très simple. Si un objet A a besoin de collaborer avec un objet B, il est nécessaire que l'objet A connaisse l'identité de B. On parle alors d'une dépendance entre les objets A et B. Cette dépendance, détectée lors de la phase d'analyse d'un projet, donne naissance en phase de conception détaillée, à l'ajout d'un attribut de type B dans la classe A.
- b) la programmation orientée aspect : permet de séparer les préoccupations transverses.
- c) une couche d'abstraction : permet d'intégrer d'autres Frameworks et bibliothèques avec une plus grande facilité.

Divers modules viennent en complément pour permettre l'intégration avec les autres bibliothèques et Frameworks, notamment :

- Spring AOP pour la programmation par aspects.
- Spring DAO pour l'accès aux données, avec un support pour JDBC, Hibernate, ...
- Spring Context et Spring MVC pour les applications web Apache Struts...
- Spring Batch pour le développement d'applications de traitement par lots.

2. Présentation de Spring Batch

Spring batch propose un Framework robuste pour le traitement par lots dans les applications d'entreprise, en se focalisant sur les grands volumes d'informations. Il permet de développer plus facilement des programmes batch en Java en répondant aux problématiques suivantes:

- Traitement «par lot»: pour éviter par exemple de charger en mémoire l'ensemble des données traitées par le batch. De plus, Spring Batch fournit des implémentations de classes permettant de lire ou d'écrire par lot sur divers types de supports (SQL, fichier plat ,etc).



➤ **Utilisation de Spring** : le développeur qui a l'habitude de Spring peut réutiliser facilement ses notions, ainsi que les composants de ce Framework tels que, par exemple, les JdbcTemplate.

3. Pourquoi Spring Batch?

Spring-Batch offre des fonctions réutilisables qui sont indispensables au traitement de gros volumes de documents.

Spring Batch est une infrastructure qui se charge de répondre à tous les besoins bas niveau dont le code est assez récurrent et répétitif. Cette couche propose ainsi une boîte à outils de composants génériques et prêts à l'emploi pour toutes les actions habituelles de manipulation de ressources (lecture de fichiers sous différents formats, gestion des connexions, ...).

Il permet aussi de normaliser et gérer le cycle de vie du batch : "quelles sont les prérequis", "par quoi commencer" et "où finir". Elle offre une architecture haut niveau avec une organisation des tâches assez précises. On parle de lancement de « jobs » qui est composé de « steps ».

4. Architecture de Spring Batch

L'architecture de ce Framework est découpée en plusieurs couches afin d'offrir un maximum de flexibilité pour son intégration dans les applications, ainsi que pour supporter différents environnements d'exécution.



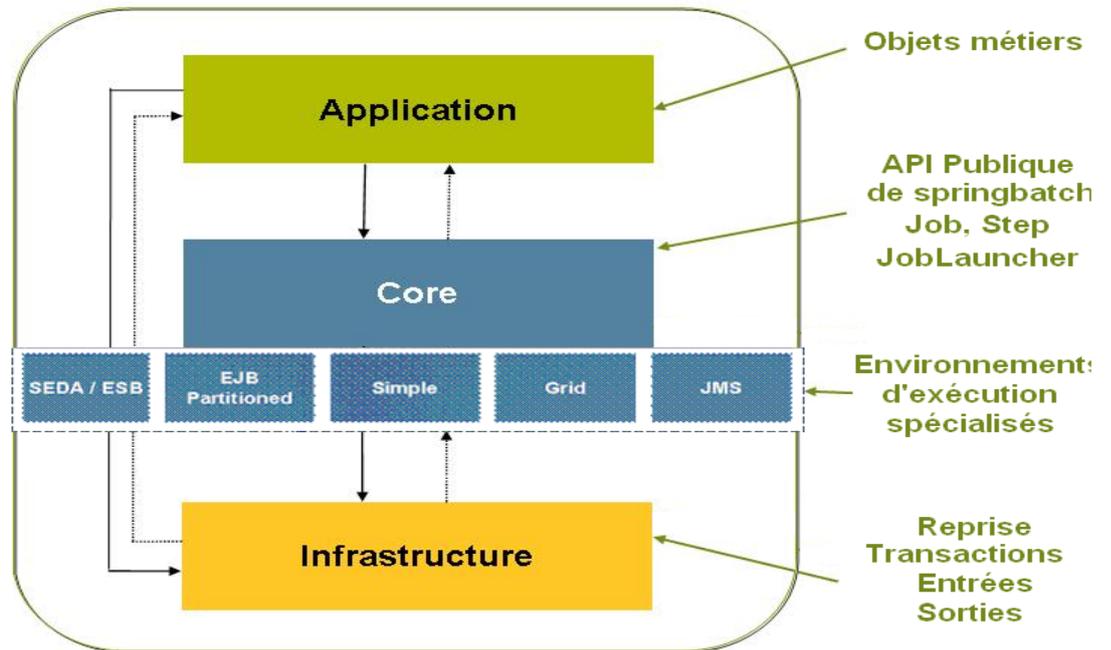


Figure 7: Architecture de Spring Batch

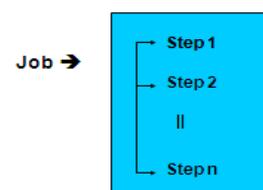
- Couche «Application» : contient tous les jobs batch et le code personnalisé écrit par les développeurs utilisant Spring Batch.
- Couche «Core»: contient l'API principale de Spring Batch et inclut tous les concepts principaux sous forme d'interfaces java, telles que les JobLauncher, Jobs, les Steps.
- Couche «Exécution»: fournit un support pour les différents environnements d'exécution des batch comme EJB, JMS.
- Couche «Infrastructure» : fournit un ensemble d'outils de bas niveau qui sont utilisés pour implémenter les couches «Exécution» et «Core» du Framework.

5. Les principaux éléments de Spring batch

a. Job

Un job correspond à un traitement qui est effectué lors de l'exécution d'un batch, sa configuration est généralement contenue dans un fichier XML et le nom du Job est déterminé par l'attribut "id".

Le Job est constitué d'un ensemble d'étapes (step). Il peut, par exemple, être constitué de trois étapes: l'une chargeant des données à



partir des fichiers pour les mettre en BD, une autre transformant ces données pour les insérer dans une autre table, et la troisième exportant les données traitées dans un fichier.

- JobInstance : La notion de job est insuffisante pour définir complètement le cycle de vie d'un batch, ce qui implique que le job doit être instancié plusieurs fois. Cependant Spring Batch introduit la notion de JobInstance pour faire référence à l'exécution d'un batch. Le JobInstance est identifié de façon unique par le couple (nom du job, liste des paramètres du job).

- JobParameters: c'est l'ensemble des paramètres utilisés pour démarrer un traitement Batch. Les paramètres passés à un job lors de son lancement permettent de définir complètement son identité, c'est-à-dire l'objet JobInstance qui le représentera. Une instance de job peut ainsi être résumée par l'équation suivante: $\text{JobInstance} = \text{Job} + \text{JobParameters}$.

- JobExecution : Les exécutions des jobs sont représentées par la classe JobExecution, elle collecte les informations relatives à une exécution d'une instance de job. Une exécution de job possède un état (status) qui peut être en cours de démarrage (STARTING), démarré (STARTED), terminé (COMPLETED), échoué (FAILED), en cours d'arrêt (STOPPING), arrêté (STOPPED) et inconnu (UNKNOWN).

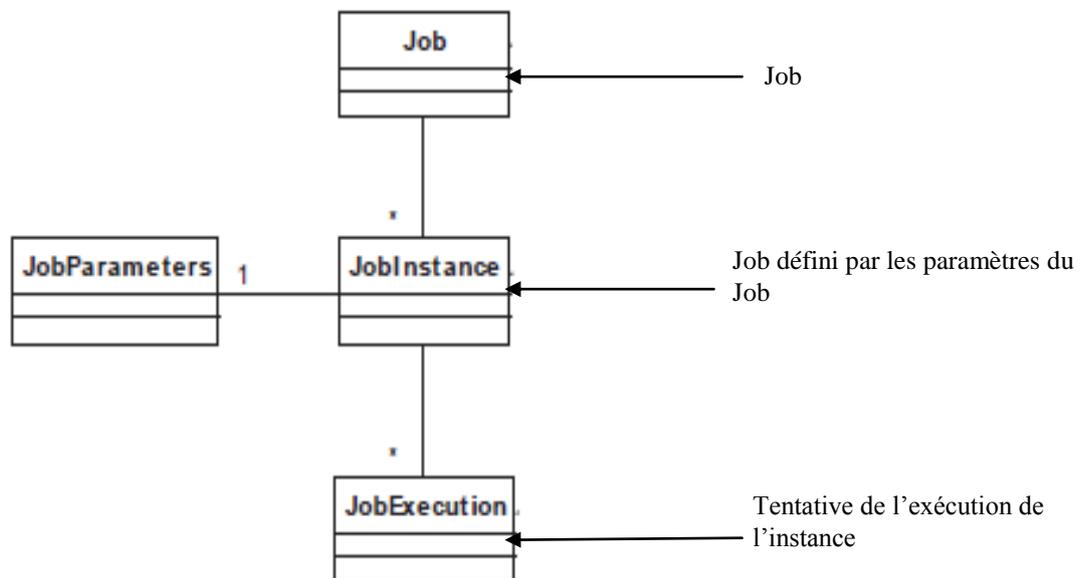


Figure 8: Cardinalités entre Job, JobInstance et JobExecution





Une étape «Step» est l'unité de traitement de base d'un job, c'est une phase séquentielle d'un travail par lots.

- StepExecution: cette classe collecte les informations relatives à une exécution d'une étape. Elle concerne une instance unique d'exécution de job (jobExecution). Une exécution d'étape possède un état (status) qui peut être en cours de démarrage (STARTING), démarré (STARTED), terminé (COMPLETED), échoué (FAILED), en cours d'arrêt (STOPPING), arrêté (STOPPED) et inconnu (UNKNOWN).

En résumé Spring Batch propose un véritable cadre de travail pour les applications utilisant des batch. Fidèle aux principes de Spring, Spring Batch propose une infrastructure gérant la plupart des problématiques techniques afin de laisser le développeur se concentrer sur le code applicatif.

B) Environnement de développement.

1. Eclipse [6]

Eclipse est un environnement de développement intégré libre, universel et polyvalent, permettant de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Il est principalement écrit en Java. La spécificité de ce dernier vient du fait de son architecture totalement développée autour de la notion de plugin.



2. Toad For Oracle [7]

Toad for Oracle est un outil puissant de développement d'application et de gestion des BD, il fournit un moyen simple et efficace pour écrire et tester des scripts et des requêtes SQL et PL/SQL, et ses puissants réseaux de données fournissent un moyen facile pour consulter et modifier des données Oracle, des procédures, des fonctions et des tables de BD.





Depuis 10 ans, Toad for Oracle reste la solution la plus avancée et la plus largement utilisée sur son marché. C'est l'outil idéal pour les experts et les nouveaux utilisateurs Oracle.

C) Langages utilisés

1. Java [8]

JAVA est un langage de programmation orientée objet qui fonctionne comme une machine virtuelle (indépendant de toute plate forme).

JAVA est très proche du langage C++ étant donné qu'il a quasiment la même syntaxe. Toutefois Java est plus simple que le langage C++, car les caractéristiques critiques du langage C++ (celles qui sont à l'origine des principales erreurs: Les pointeurs, L'héritage multiple et surcharge d'opérateurs) ont été supprimées.

2. SQL/PL SQL [9]

SQL (*Structured Query Language*) est un langage informatique créé par IBM pour l'exploitation des BD. Les systèmes de gestion de BD qui sont apparus par la suite ont tous pris en charge ce langage, ce qui a fini par en faire le langage standard d'interrogation, normalisé par l'ANSI. Ce langage sert à effectuer des opérations sur les BD, il permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les BD.

PL/SQL (*Procedural Language / Structured Query Language*) est un langage procédural qui permet de combiner des requêtes SQL et des instructions procédurales (boucles, conditions...).

Conclusion:

L'utilisation de cet environnement technique va permettre de créer un projet solide et bien structuré qui facilitera le travail du développeur des promotions fixes de MT.





Chapitre 3 :

Etude de la BD de MT

Ce chapitre est consacré à l'étude des tables utilisées dans la création de l'API.

Introduction:

MT continue à profiter du prodigieux potentiel du secteur des télécoms au Maroc qui ne montre aucun signe d'essoufflement jusqu'à aujourd'hui, malgré la concurrence qui y fait rage depuis des années.

En raison de ce potentiel, MT a mis à sa disposition une large BD propre à son segment, permettant mieux d'organiser et de structurer la gestion de son réseau, de supporter ses activités comme la gestion des commandes et des clients, d'augmenter les possibilités de diffuser et d'accéder à l'information d'une manière rapide et améliorée. Cette BD est constituée de 1005 tables.



Pour pouvoir réaliser mon projet, j'ai fait une étude sur 25 tables, afin de comprendre le contenu et l'utilité de chaque table, la relation entre elles et enfin les différents champs. Ces tables sont nécessaires et toujours utilisées pour développer toutes les promotions fixes de MT.

Ces tables sont généralement réparties en tables pour les clients, les contrats, les factures, les services, les PT, et les promotions. La relation entre ces tables sera modélisée par le langage UML.

I. Analyse et conception

UML (Unified Modeling Language ou langage de modélisation unifié) est un langage de modélisation normalisé développé par l'OMG (Object Management Group).

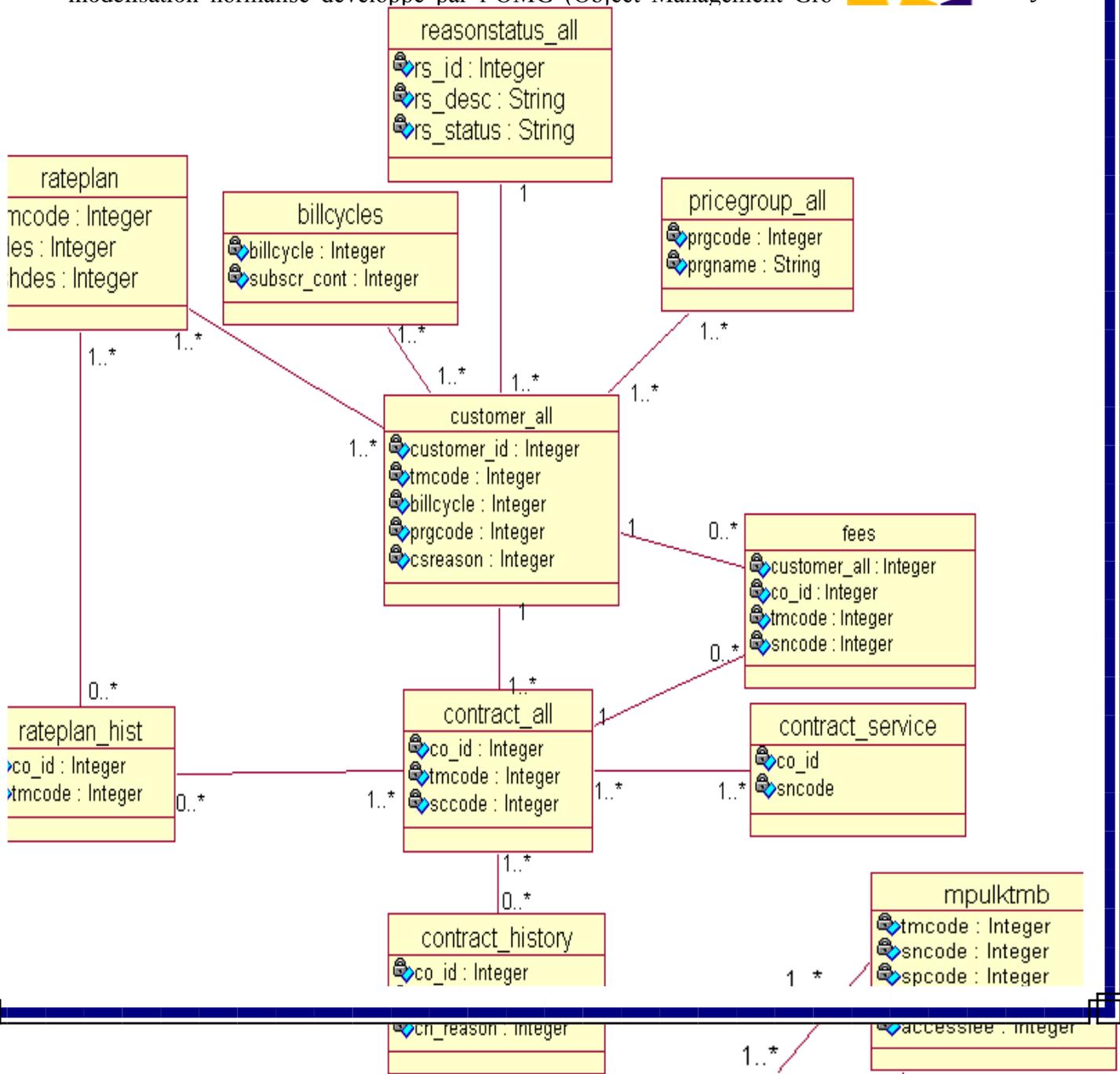


Figure 9: Diagramme de classes client/contrat

b. Diagramme contrat / service

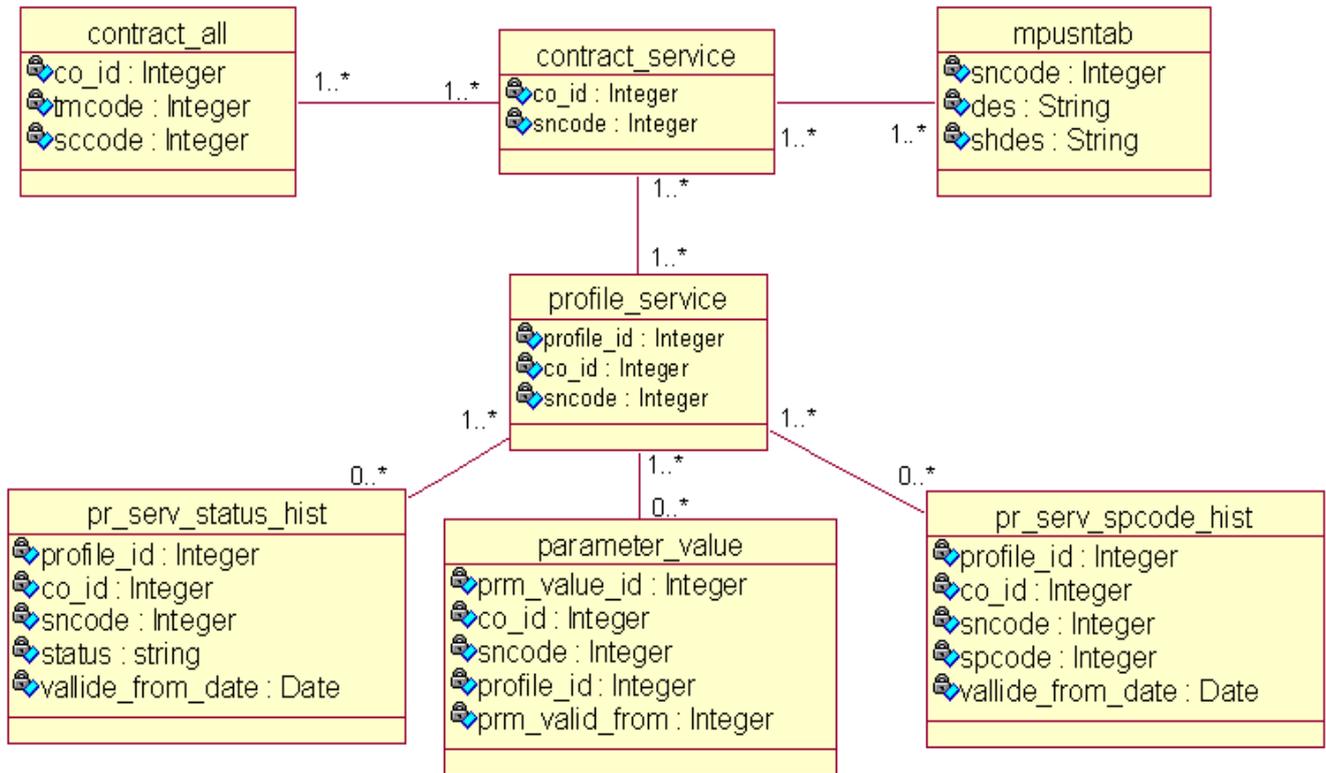


Figure 10: Diagramme de classes contrat/service

c. Diagramme factures

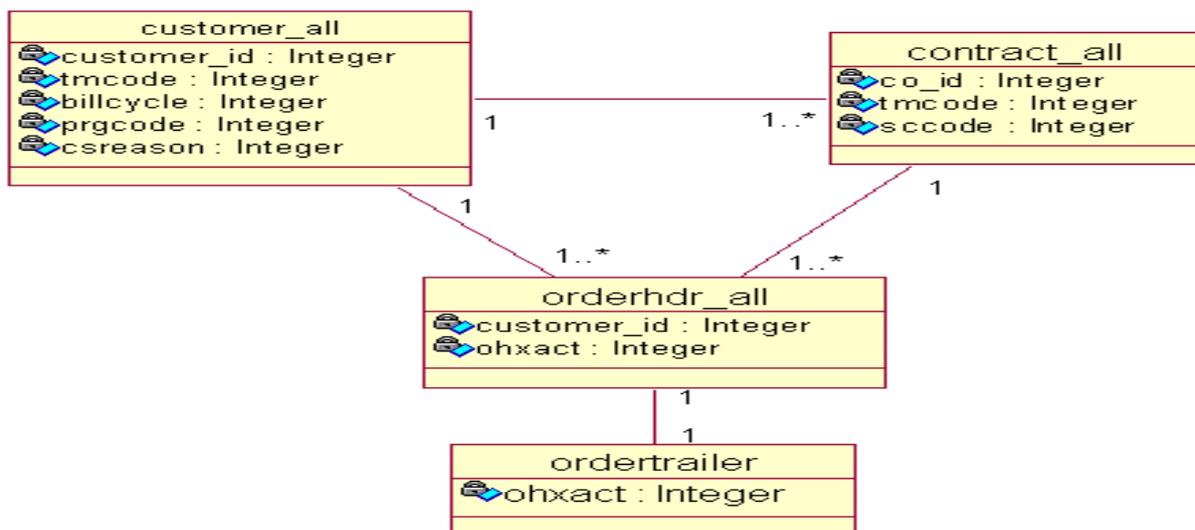


Figure 11 : Diagramme de classes factures.



d. Diagramme des promotions

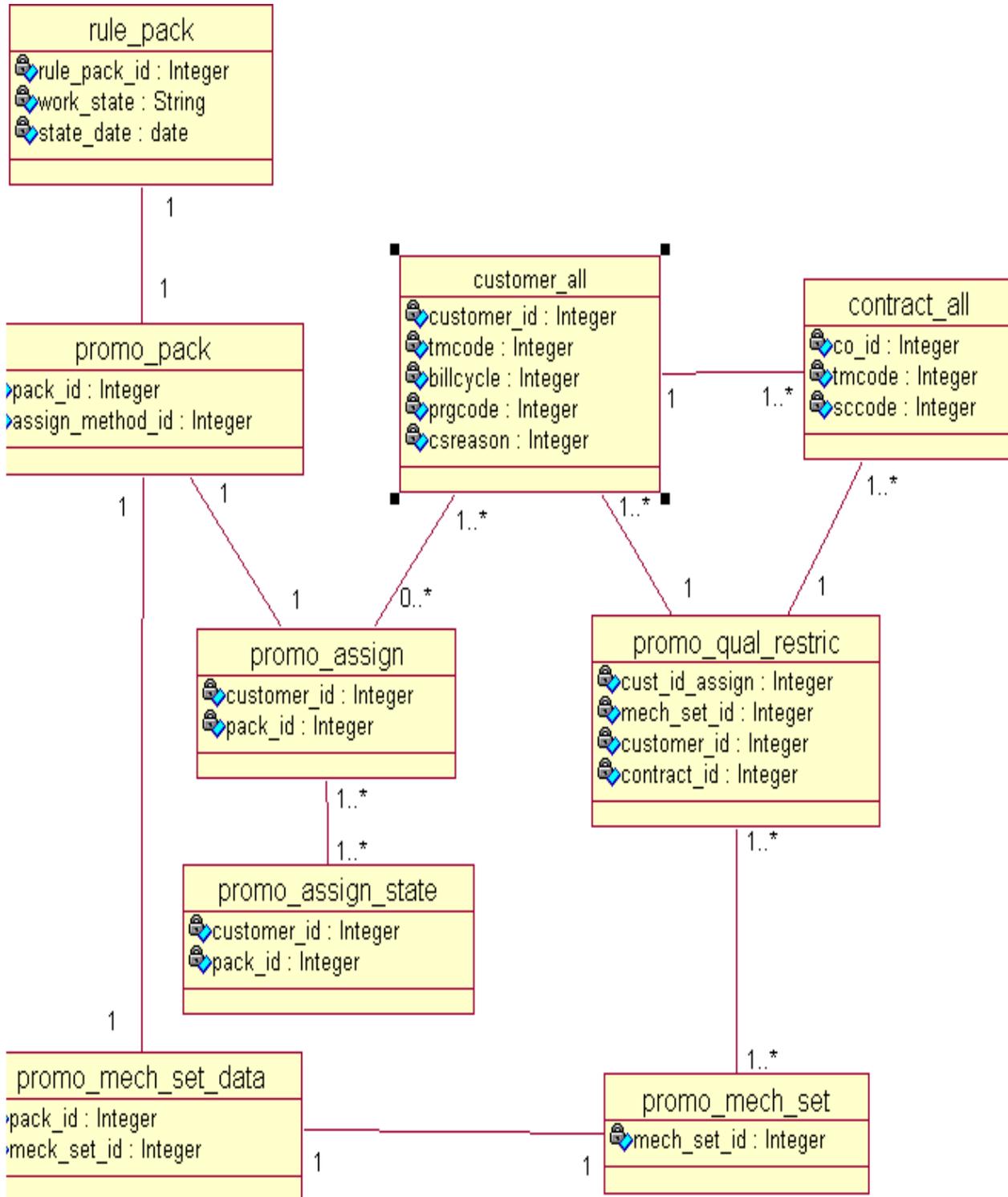


Figure 12: Diagramme de classes promotions.



3. Caractéristiques des tables

Chaque table contient plusieurs champs. Les tableaux suivants montrent quelques caractéristiques, de chaque table, qui sont utilisées dans l'application.

- Table CUSTOMER_ALL: table des clients.

Elle contient les attributs suivants :

Nom du champ	Format de données	Domaine de valeur	Description
Customer_id	Numérique		Numéro du client
Tmcode	Numérique	Défini dans la table RATEPLAN	Code PT
Prgcode	Numérique	Défini dans la table PRICEGROUP_ALL	Code de la catégorie
Bilcycle	Numérique	Défini dans la table BILLCYCLES	Numéro de chaque cycle de facturation
Csreason	Numérique		Code du motif du changement de la situation

Tableau 1: Attributs de la classe CUSTOMER_ALL

- Table PRICEGROUP_ALL: table des catégories des clients.

Elle contient les attributs suivants :

Nom du champ	Format de données	Domaine de valeur	Description
Prgcode	Numérique		Code de la catégorie
Prgname	Alphabétique (30BYTE)		Nom de la catégorie

Tableau 2: Attributs de la classe PRICEGROUP_ALL

- Table BICYCLES : table des cycles de facturation.

Elle contient les attributs suivants :

Nom du champ*	Format de données	Domaine de valeur	Description
Bilcycle	Numérique		Numéro de chaque cycle de facturation
Subscr_cont	Numérique		nombre d'abonnés contenus dans ce cycle de facturation

Tableau 3: Attributs de la classe BICYCLES

- Table REASONSTATUS_ALL : Cette table définit la raison du changement du statut de l'abonné, elle contient les attributs suivants:





Nom du champ*	Format de données	Domaine de valeur	Description
Rs_id	Numérique		Code de la raison du changement du statu de l'abonné
Rs_desc	Alphabétique (40 byte)		Description de la raison du changement
Rs_status	Alphabétique (1byte)		Courte description

Tableau 4: Attributs de la classe REASONSTATUS_ALL

- Table RATEPLAN: table des PT.

Elle contient les attributs suivants:

Nom du champ*	Format de données	Domaine de valeur	Description
Tmcode	Numérique		Code du PT
Des	Alphabétique (30 Byte)		Description du PT
Shdes	Alphabétique (5 Byte)		Courte description du PT

Tableau 5: Attributs de la classe RATEPLAN

- Table RATEPLAN_HIST : historique des PT par contrat.

Elle contient les attributs suivants :

Nom du champ*	Format de données	Domaine de valeur	Description
Co_id	Numérique	Défini dans la table CONTRACT_ALL	Numéro du contrat
tmcode	Numérique	Défini dans la table RATEPLAN	Code du PT
Tmcode_date	Date		Date depuis laquelle le tmcode est validé
Userlastmod	Alphabétique (16 byte)		L'utilisateur qui a effectué le changement du PT

Tableau 6: Attributs de la classe RATEPLAN_HIST

- Table FEES : table d'insertion des OCC pour les clients.

Elle contient les attributs suivants :

Nom du champ*	Format de données	Domaine de valeur	Description
Customer_id	Numérique	Défini dans la table CUSTOMER_ALL	Numéro du client



Co_id	Numérique	Défini dans la table CONTRAT_ALL	Numéro du contrat
Amount	Numérique		Les frais d'OCC à appliquer
Tmcode	Numérique	Défini dans la table RATEPLAN	Code du PT
Sncode	Numérique	Défini dans la table MPUSNTAB	Code du service
username	Alphabétique (16byte)		Nom d'utilisateur BSCS
Valid_from	Date		Date à laquelle les frais d'OCC seront facturés

Tableau 7: Attributs de la classe FEES

- Table CONTRACT_ALL : table des contrats.

Elle contient les attributs suivants :

Nom du champ*	Format de données	Domaine de valeur	Description
Co_id	Numérique		Numéro du contrat
Tmcode	Numérique	Défini dans la table RATEPLAN	Code du PT
Sccode	Numérique		Identificateur du marché (GSM, Fixe)
Co_signed	Date		Date à laquelle le contrat a été signé
Co_entdate	Date		Date de la première activation du contrat
Co_activated	Date		Date d'annulation du contrat

Tableau 8: Attributs de la classe CONTRACT_ALL

- Table CONTRACT_HISTORY : historique des statuts des contrats.

Elle contient les attributs suivants :

Nom du champ*	Format de données	Domaine de valeur	Description
Co_id	Numérique	Défini dans la table CONTRACT_ALL	Numéro du contrat
Ch_sqno	Numérique		numéro consécutif incrémenté de 1 pour chaque changement de statut



Ch_status	Alphabétique (1BYTE)		valeurs d'état possibles sont: o: en attente (doit être le premier statut) a: activé s: suspendu d: désactivé
Ch_reason	Numérique		Code de la raison du changement d'état
Co_entdate	Dtae		Date de la première activation

Tableau 2: Attributs de la classe CONTRACT_HISTORY

- Table CONTRACT_SERVICE : table des services par contrat.

C'est la table lien entre les contrats et les services, elle contient les attributs suivants:

Nom du champ*	Format de données	Domaine de valeur	Description
Co_id	Numérique	Défini dans la table CONTRACT_ALL	Numéro du contrat
Sncode	Numérique	Défini dans la table MPUSNTAB	Code du service

- **Tableau 10: Attributs de la classe CONTRACT_SERVICE**

- Table MPUSNTAB : table des services.

Elle contient les attributs suivants :

Nom du champ*	Format de données	Valeur	Description
Sncode	Numérique		Code du service
shdes	Alphabétique (5BYTE)		Courte description de sncode
des	Alphabétique(30BYTE)		Description de sncode

Tableau 11: Attributs de la classe MPUSNTAB

- Table PROFILE_SERVICE : table des services par profile.

Elle contient les attributs suivants

Nom du champ*	Format de données	Domaine de valeur	Description
Profile_id	Numérique		Numéro du profile
Co_id	Numérique	Défini dans la table CONTRACT_ALL	Numéro du contrat
Sncode	Numérique	Défini dans la table MPUSNTAB	Code du service
Ovw_adv_charge	Alphabétique (1byte)		La manière d'attribution d'OVW. A: absolue R: relative
Adv_charge	Numérique		il s'agit de l'avance à facturer aux



			clients.
Adv_charge_prd	Numérique		Combien de fois il faut appliquer l'ADV_CHARGE

Tableau 12: Attributs de la classe PROFILE_SERVICE

- Table PR_SERV_STATUS_HIST: historique des statuts des services par profile.

Elle contient les attributs suivants:

Nom du champ*	Format de données	Domaine de valeur	Description
Profile_id	Numérique	Défini dans la table PROFILE_SERVICE	Numéro du profile
Co_id	Numérique	Défini dans la table CONTRACT_ALL	Numéro du contrat
Sncode	Numérique	Défini dans la table MPUSNTAB	Code du service
status	Alphabétique (1byte)		Statut du service dans un profile A : activé D : désactivé S : suspendu
Reason	Numérique		Code de la raison du changement de statut par ex : 3 : changement du PT 9 : service invisible
Valid_from_date	Date		Date depuis laquelle le statut est validé

Tableau 13: Attributs de la classe PR_SERV_STATUS_HIST

- Table PR_SERV_SPCODE_HIST: historique des packages des services. Elle contient les attributs suivants :

Nom du champ*	Format de données	Domaine de valeur	Description
Profile_id	Numérique	Défini dans la table PROFILE_SERVICE	Numéro du profile
Co_id	Numérique	Défini dans la table CONTRACT_ALL	Numéro du contrat
Sncode	Numérique	Défini dans la table MPUSNTAB	Code du service
spcode	Numérique		Code du package du service

Tableau 14: Attributs de la classe PR_SERV_SPCODE_HIST

- Table PARAMETER_VALUE : historique des paramètres par service.

Elle contient les attributs suivants :

Nom du champ*	Format de données	Domaine de valeur	Description
Prm_value_id	Numérique		Identificateur de la valeur du paramètre



Co_id	Numérique	Défini dans la table CONTRACT_ALL	Numéro du contrat
Sncode	Numérique	Défini dans la table MPUSNTAB	Code du service
Profile_id	Numérique	Défini dans la table PROFILE_SERVICE	Numéro du profile
Parameter_id	Numérique		Identificateur du paramètre
Prm_description	Alphabétique (100byte)		Description du paramètre
Prm_valid_from	Date		La date d'activation du paramètre

Tableau 15: Attributs de la classe PARAMETER_VALUE

- Table ORDERHDR_ALL : table des factures par client.

Elle contient les attributs suivants :

Nom du champ*	Format de données	Domaine de valeur	Description
Ohxact	Numérique		numéro de la commande
Ohendate	Date		date de la saisie de la commande
Ohipp	Alphabétique (6BYTE)		période d'affichage de la facture
Customer_id	Numérique	Défini dans la table CUSTOMER_ALL	Numéro du client

Tableau 16: Attributs de la classe ORDERHDR_ALL

- Table ORDERTRAILER: détail de chaque facture.

Elle contient les attributs suivants :

Nom du champ*	Format de données	Domaine de valeur	Description
Ohxact	Numérique	Défini dans la table ORDERHDR_ALL	numéro de la commande
Qtqord	Numérique		Quantité commandée
Otqbkord	Numérique		Quantité de rupture de stock
Otqtoshp	Numérique		Quantité à expédier
Otjobcost	Numérique		Code de cout de revient
Servcat_code	Alphabétique (2BYTE)		Code de la catégorie du service

Tableau 17: Attributs de la classe ORDERTRAILER

- Table MPULKTMB: table lien entre les PT et les services packages.

Elle contient les attributs suivants :

Nom du champ*	Format de données	Domaine de valeur	Description
Tmcode	Numérique	Défini dans la table RATEPLAN	Code du PT
Sncode	Numérique	Défini dans la table	Code du service



		MPUSNTAB	
Spcode	Numérique		Code du package du service
subscript	Numérique		Frais d'inscription
Accessfee	Numérique		Frais d'abonnement
Echnid	Alphabétique (1byte)		Manière d'attribution des frais A = Absolu, R = Relative
Amtind	Alphabétique (1byte)		Montant F = fixe, C = Modifiable
Frqind	Alphabétique (1byte)		Fréquence O = une seule fois, P = périodique, I =intervalle

Tableau 18: Attributs de la classe MPULKTMB

Tables des promotions :

- Table RULE_PACK: description des règles du package de la promotion.

Elle contient les attributs suivants:

Nom du champ*	Format de données	Domaine de valeur	Description
Rule_pack_id	Numérique	Défini dans la table PROMO_PACK	Numéro du package
Des	Alphabétique (35 byte)		description du package
shdes	Alphabétique (5 byte)		Courte description du package
Work_state	Alphabétique (1byte)		L'état réel de fonctionnement de ce package. Les états possibles sont: P = dans la production W = encore sous le travail
State_date	Date		Date depuis laquelle l'état a été fixé

Tableau 19: Attributs de la classe RULE_PACK

- Table PROMO_PACK : définit le package de la promotion.

Elle contient les attributs suivants :

Nom du champ*	Format de données	Domaine de valeur	Description
Pack_id	Numérique		Numéro du package
Assign_method_id	Numérique		Numéro du code de la méthode d'affectation de la promotion
Avail_from	Date		Date de début de disponibilité de la promotion
Avail_to	Date		Date de fin de disponibilité

Tableau 20: Attributs de la classe PROMO_PACK

- Table PROMO_ASSIGN: table d'assignement de la promotion.



Elle contient les attributs suivants:

Nom du champ*	Format de données	Domaine de valeur	Description
Customer_id	Numérique	Défini dans la table CUSTOMER_ALL	Numéro du client
Pack_id	Numérique	Défini dans la table PROMO_PACK	Numéro du package
Assign_date	Date		Date de l'attribution de la promotion
Delete_date	Date		Date de la suppression de cette promotion

Tableau 21: Attributs de la classe PROMO_ASSIGN

- Table PROMO_ASSIGN_STATE : l'état de la promotion.

Elle contient les attributs suivants.

Nom du champ*	Format de données	Domaine de valeur	Description
Customer_id	Numérique	Défini dans la table CUSTOMER_ALL	Numéro du client
Pack_id	Numérique	Défini dans la table PROMO_PACK	Numéro du package
Work_state	Alphabétique (1byte)		Etat actuel de la promotion: A: active/I: inactive
State_date	Date		Date de la création de l'état

Tableau 22: Attributs de la classe PROMO_ASSIGN_STATE

- Table PROMO_MECH_SET : contient le mécanisme de la promotion.

Elle contient les attributs suivants :

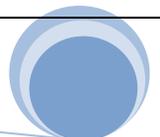
Nom du champ*	Format de données	Domaine de valeur	Description
Mech_set_id	Numérique		Identificateur du mécanisme de la promotion
Short_description	Alphabétique(5 byte)		Courte description du mécanisme
Description	Alphabétique(30 byte)		Description du mécanisme

Tableau 23: Attributs de la classe PROMO_MECH_SET

- PRMO_MECH_SET_DATA: table lien entre le mécanisme et le package de la promotion. Elle contient les attributs suivants :

Nom du champ*	Format de données	Domaine de valeur	Description
Mech_set_id	Numérique	Défini dans la table PRMO_MECH_SET	Identificateur du mécanisme de la promotion
Pack_id	Numérique	Défini dans la table RULE_PACK	Numéro du package

Tableau 24: Attributs de la classe PROMO_MECH_SET_DATA



- Table PROMO_QUAL_RESTRICT: table des restrictions pour les clients.

Elle contient les attributs suivants.

Nom du champ*	Format de données	Domaine de valeur	Description
Cust_id_assign	Numérique		Identificateur du client qui lui a été assignée la promotion
Mech_set_id	Numérique	Défini dans la table PROMO_MECH_SET	Identificateur du mécanisme de la promotion
Customer_id	Numérique	Défini dans la table CUSTOMER_ALL	Numéro du client
Contract_id	Numérique	Défini dans la table CONTRACT_ALL	Numéro du contrat

Tableau 25: Attributs de la classe PROMO_QUAL_RESTRICT

Conclusion :

MT dispose d'une large BD qui organise et structure son réseau. La réalisation du projet a nécessité une étude sur 25 tables qui feront la base pour le développement de l'application.



Chapitre 4 :

Mise en œuvre

Ce chapitre présente la partie réalisation de l'API et une description des méthodes utilisées pour sa réalisation ainsi que l'atelier utilisé. Ensuite des prises d'écrans seront présentées comme démonstration.

III. Description du projet

Après avoir étudié les différentes tables nécessaires pour la mise en œuvre de l'application, j'ai commencé la réalisation de la partie factorisation en passant par deux étapes.

La première étape était réalisée avec Toad For Oracle pour exécuter les requêtes indispensables pour le développement des promotions fixes.

La deuxième étape était réalisée sous Eclipse :

- Création d'un fichier XML « req.xml » qui contient l'ensemble des requêtes exécutées sous Toad For Oracle.
- Création d'une classe java contenant les fonctions set et get de chaque requête incluse sur req.xml.
- Création d'une classe qui contient les méthodes qui font appel aux requêtes.
- Exécution du projet.





1. Description des méthodes développées

1-GetSysdate

Description : cette méthode permet de retourner la date du system selon le format demandé. Il faut prévoir dans la méthode que le format est un paramètre en entrée==> un contrôle doit être fait sur tous les formats possibles :

Ex : DD-MM-YYYY, DD/MM/AA, DD/MM/YYYY

Arguments en entrée :

Nom argument	Type	Exemple de valeur
p_format	Date	DD/MM/AAAA

Type : fonction.

Retour : la date du system si le format est correct, sinon la valeur KO (message : la valeur n'existe pas ou erreur).

2- GetSncode

Description : La méthode doit retourner le code d'un service à partir de son SHDES

Arguments en entrée :

Nom argument	Type	Longueur	Exemple de valeur
SHDES	Alphabétique	5	EB058

Type : Fonction.

Retour : la valeur du SNCODE du service si c'est correct, sinon la valeur KO.

3- GetTmcode

Description : Cette méthode permet d'avoir le code d'un PT à partir de son SHDES.

Arguments en entrée :

Nom argument	Type	Longueur	Exemple de valeur
SHDES	Alphabétique	5	LLN15

Type : fonction.

Retour : la valeur du TMCODE du PT Si c'est correct, sinon la valeur KO.

4- GetPackID

Description : La méthode doit retourner le numéro du package d'un service à partir de son short_description.

Arguments en entrée :

Nom argument	Type	Longueur	Exemple de valeur
SHORT_DESCRIPTION	Alphabétique	5	PA006

Type : Fonction.

Retour : la valeur du RULE_PACK_ID du service si correct, sinon la valeur KO.

5-GetCatCustomer

Description : la méthode doit retourner le code de la catégorie d'un client à partir de son numéro.

Arguments en entrée :





Nom argument	Type	Exemple de valeur
CUSTOMER_ID	Numérique	3073

Type : Fonction.

Retour : la valeur du PRGCODE du client si c'est correct, sinon la valeur KO.

6 – GetContractInfo

Description : La méthode doit retourner les informations sur un contrat donné.

Arguments en entrée :

Nom argument	Type	Exemple de valeur
CO_ID	Numérique	28663

Type : Fonction.

Retour : les informations significatives pour un contrat comme TMCODE, CO_SIGNED, CO_ACTIVATED si c'est correct, sinon la valeur KO.

7 – GetLastContractRp

Description : La méthode doit retourner le dernier PT d'un contrat et le numéro du client avant une date donnée.

Arguments en entrée :

Nom argument	Type	Exemple de valeur
CO_ID	Numérique	28513
TMCODE_DATE	Date	01/08/2004

Type : Fonction.

Retour : la valeur du dernier TMCODE d'un contrat et le CUSTOMER_ID si c'est correct, sinon la valeur KO.

8 – GetSrvParam

Description : La méthode doit retourner le dernier paramètre d'un service d'un contrat donné.

Arguments en entrée :

Nom argument	Type	Exemple de valeur
CO_ID	Numérique	28724
SNCODE	Numérique	211
PARAMETER_ID	Numérique	34

Type : Fonction.

Retour : La valeur PRM_DESCRIPTION si c'est correct, sinon la valeur KO.

9 – GetParamValueInDate

Description : La méthode doit retourner la valeur du paramètre d'un service d'un contrat donné avant une date donnée.

Arguments en entrée :

Nom Argument	Type	Exemple de valeur
CO_ID	Numérique	28724
PRM_VALID_FROM	Date	29/01/2005



Type : Fonction.

Retour : La valeur PRM_VALUE_ID donnée si c'est correct, sinon la valeur KO.

10 - GetSrvStatus

Description : La méthode doit retourner le statut d'un service d'un contrat avant une date donnée en paramètre.

Arguments en entrée :

Nom Argument	Type	Longueur	Exemple Valeur
CO_ID	Numérique		28744
SHDES	Numérique	5	EB058
SNCODE	Numérique		105
VALID_FROM_DATE	Date		27/08/2007

Type : Fonction.

Retour : La valeur STATUS d'un service si c'est correct, sinon la valeur KO.

11- GetLastSrvStatus

Description : La méthode doit retourner le dernier statut d'un service d'un contrat et la date de sa validation.

Arguments en entrée :

Nom argument	Type	Longueur	Exemple de valeur
CO_ID	Numérique		28744
SHDES	Numérique	5	EB058

Type : Fonction.

Retour : La dernière valeur de STATUS et la valeur VALID_FROM_DATE d'un service si c'est correct, sinon la valeur KO.

12- AssignPromo:

Description : La méthode doit appliquer une promotion pour un client donné.

Il y'a deux étapes à faire:

- à partir du SHORT_DESCRIPTION, on récupère le RULE_PACK_ID.
- avec le RULE_PACK_ID récupéré, on applique la promotion au client.

Arguments en entrée :

Nom argument	Type	Longueur	Exemple de valeur
CUSTOMER_ID	Numérique		5007701
SHORT_DESCRIPTION	Alphabétique	5	PA102
PACK_ID	Numérique		25
ASSIGN_DATE	Date		04/10/2011

Type : procédure.

13 – AssignPromoState

Description : La méthode doit appliquer une promotion pour un client donné en spécifiant la date d'effet de la promotion.

Il y'a deux étapes à faire:

- à partir du SHORT_DESCRIPTION, on récupère le RULE_PACK_ID.
- avec le RULE_PACK_ID récupéré, on applique la promotion au client.





Arguments en entrée :

Nom argument	Type	Longueur	Exemple de valeur
CUSTOMER_ID	Numérique		3073
SHORT_DESCRIPTION	Alphabétique	5	PA102
PACK_ID	Numérique		24
WORK_STATE	Alphabétique	1	A
STATE_DATE	Date		04/11/2011

Type : procédure.

14 – CheckPromoAssigned

Description : La méthode doit vérifier si une promotion est assignée ou non pour un client donné.

Arguments en entrée :

Nom argument	Type	Longueur	Exemple de valeur
PCUSTOMER_ID	Numérique		5007701
PACK_ID	Numérique		11
SHORT_DESCRIPTION	Date	5	PA102

Type : Fonction.

Retour : La méthode retourne 1 si la promotion est déjà assignée au client, KO si la promotion n'est pas assignée au client.

15 – InsertOcc.

Description : La méthode doit insérer une OCC dans la table FEES pour un client donné, contrat donné et un service donné.

Arguments en entrée :

Nom argument	Type	Exemple de valeur
CUSTOMER_ID	Numérique	3115
CO_ID	Numérique	26266
SNCODE	Numérique	444

Type : procédure.

16 – SetOverWriteSrv

Description : La méthode doit appliquer un OVW sur un service pour un contrat donné.

Arguments en entrée :

Nom argument	Type	Longueur	Exemple de valeur
CO_ID	Numérique		28724
SNCODE	Numérique		272
ADV_CHARGE	Numérique		0.996765
OVW_ADV_CHARGE	Alphabétique	1	A
ADV_CHARGE_PRD	Numérique		1

Type : procédure.

17 - GetOverWriteInfo:

Description : La méthode doit retourner les informations de d'OVW d'un service pour un contrat donné.

Arguments en entrée :





Nom argument	Type	Longueur	Exemple de valeur
CO_ID	Numérique		28293
SNCODE	Numérique		211

Type : fonction.

Retour : Les valeurs OVW_ADV_CHARGE, ADV_CHARGE, ADV_CHARGE_ PRD si c'est correct, sinon la valeur KO.

18 – GetSubscriptionAmt :

Description : La méthode doit retourner le montant d'accès à un service donné.

Arguments en entrée :

Nom argument	Type	Exemple de valeur
TMCODE	Numérique	19
SNCODE	Numérique	272

Type : fonction.

Retour : La valeur SUBSCRIPT si c'est correct, sinon la valeur KO.

19 - GetAccesAmt :

Description : La méthode doit retourner le montant de l'abonnement à un service donné.

Arguments en entrée :

Nom argument	Type	Exemple de valeur
TMCODE	Numérique	35
SNCODE	Numérique	203

Type : fonction.

Retour : La valeur ACCESSFEE si c'est correct, sinon la valeur KO.

2. Atelier

Afin de pouvoir exécuter l'application, il était nécessaire de respecter la structure d'un projet batch.

a. Description des packages: Le répertoire src/main/java

Le répertoire *src/main/java* est structuré par des packages, avec le contenu suivant :

- **advice**: contient des class génériques pour le system Log.
- **dao**: contient les classes qui ont des accès sur les objets donnés, ainsi que la classe qui fait appel aux différentes requêtes citées au dessus.
- **domaine.conf**: contient la classe de configuration « Conf » utilisé par le batch qui contient les paramètres utilisés par les différentes step du batch.
- **Lunch**: contient la classe pour lancer et exécuter l'application.





- **requetes**: contient des classes génériques pour la lecture des requêtes à partir d'un fichier XML « req.xml ».
- **Tasklet**: contient la classe pour le traitement d'une step.
- **Util**: contient la classe Trace.

b. Description des ressources :le répertoire src/main/resources

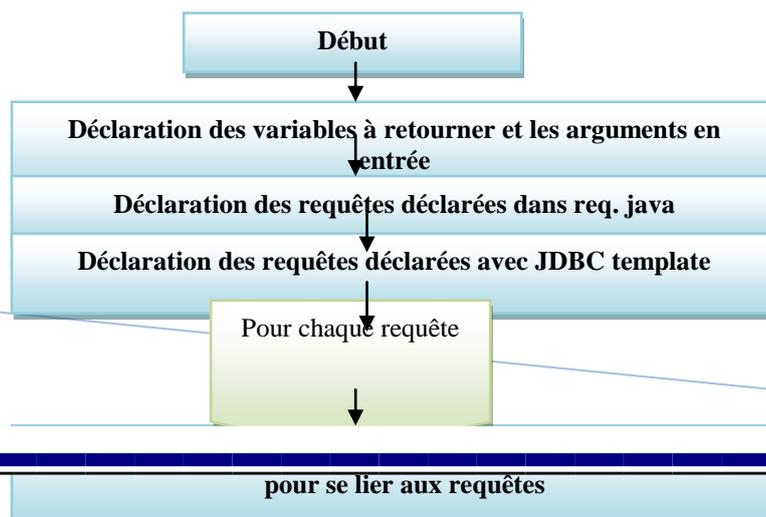
Ce répertoire est destiné à accueillir les ressources nécessaires à l'application. Il contient:

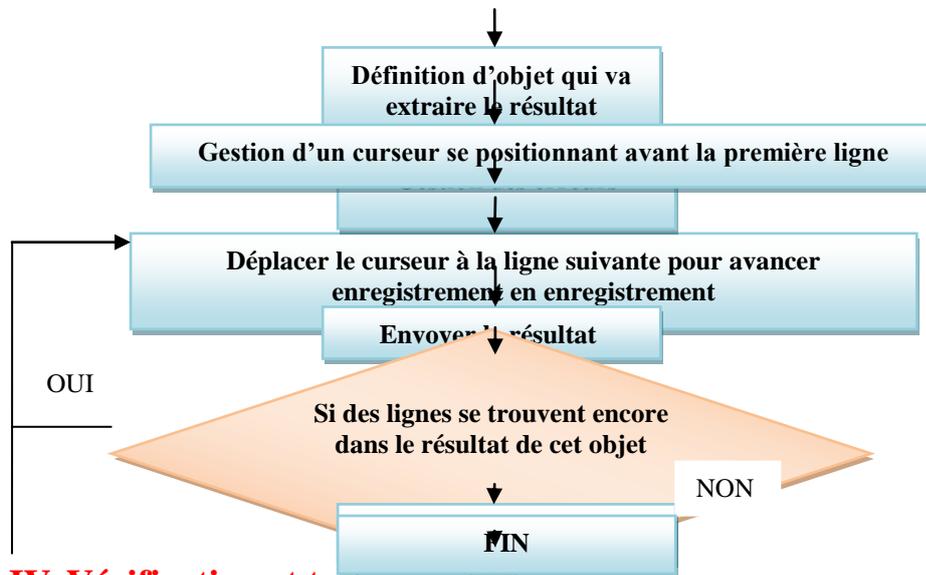
- **batch-oracle.properties**« configuration d'accès à la base»: c'est un fichier de configuration qui contient les paramètres d'accès à la base (URL de la base, utilisateur de la base, mot de passe de la base, les paramètres du driver oracle).
- **Log4j.properties**: c'est la configuration du système log.
- **req.xml**: descripteur des requêtes du batch.

c. Exécution du batch

- **BatchLauncher** : C'est la classe main du programme java qui se charge du lancement du batch.
- **LireProperties** : cette classe permet de lire le contenu d'un fichier de configuration.

3. Organigramme de la classe qui fait appel aux requêtes.



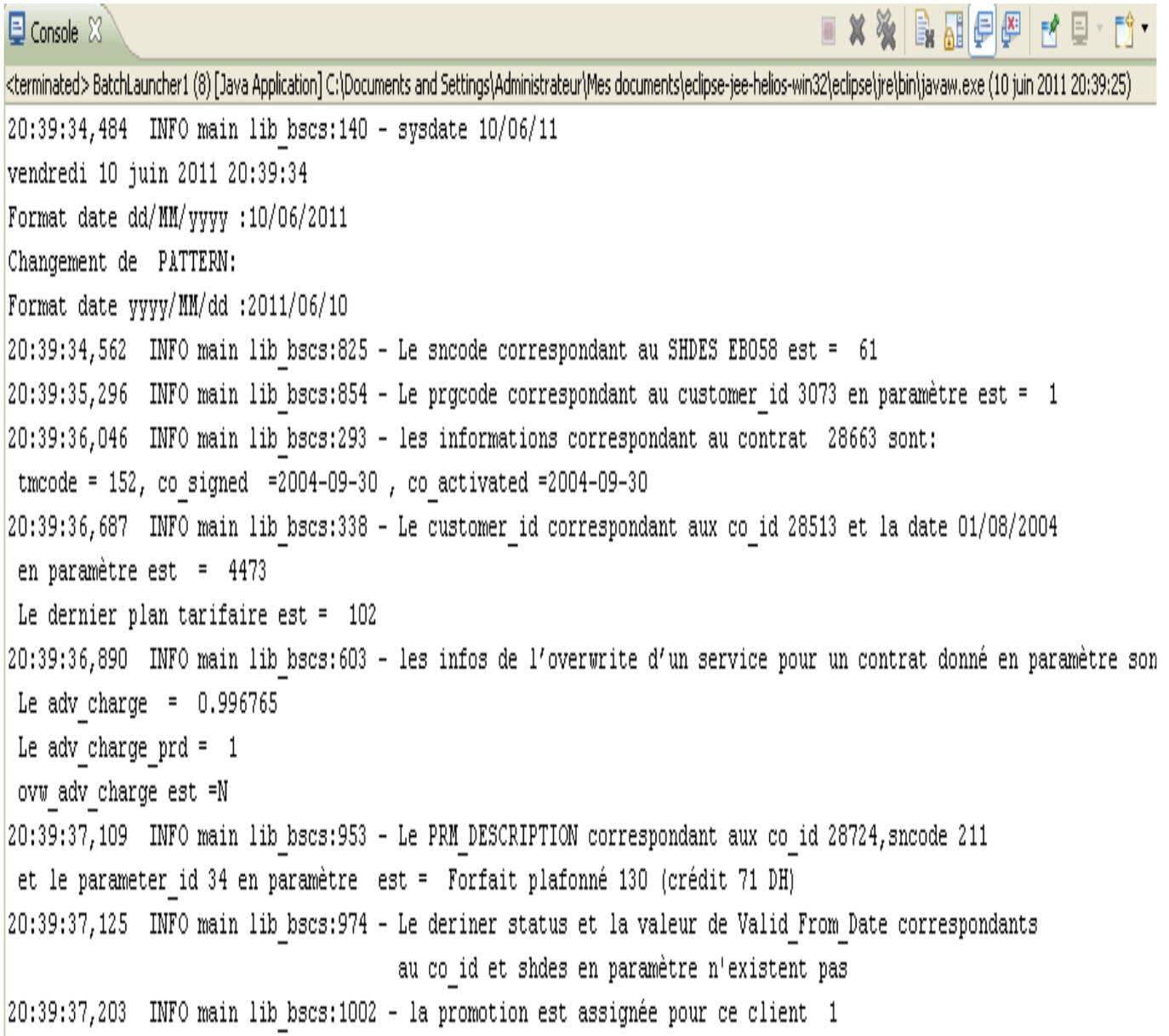


IV. Vérification et test

Comme précisé précédemment, je vais présenter quelques prises d'écrans, d'Eclipse et de Toad ForOracle, comme exemple de démonstration de l'application.

La prise d'écran suivante, obtenue avec Eclipse, présente le résultat obtenu de quelques méthodes citées au dessus, ainsi que la méthode qui retourne la date system selon le format demandé. Ces tests ont été effectués le vendredi 10 juin 2011 à 20:39:34.





```
<terminated> BatchLauncher1 (8) [Java Application] C:\Documents and Settings\Administrateur\Mes documents\eclipse-jeelios-win32\eclipse\jre\bin\javaw.exe (10 juin 2011 20:39:25)
20:39:34,484 INFO main lib_bscs:140 - sysdate 10/06/11
vendredi 10 juin 2011 20:39:34
Format date dd/MM/yyyy :10/06/2011
Changement de PATTERN:
Format date yyyy/MM/dd :2011/06/10
20:39:34,562 INFO main lib_bscs:825 - Le sncode correspondant au SHDES EB058 est = 61
20:39:35,296 INFO main lib_bscs:854 - Le prgcode correspondant au customer_id 3073 en paramètre est = 1
20:39:36,046 INFO main lib_bscs:293 - les informations correspondant au contrat 28663 sont:
tmcode = 152, co_signed =2004-09-30 , co_activated =2004-09-30
20:39:36,687 INFO main lib_bscs:338 - Le customer_id correspondant aux co_id 28513 et la date 01/08/2004
en paramètre est = 4473
Le dernier plan tarifaire est = 102
20:39:36,890 INFO main lib_bscs:603 - les infos de l'overwrite d'un service pour un contrat donné en paramètre son
Le adv_charge = 0.996765
Le adv_charge_prd = 1
ovw_adv_charge est =N
20:39:37,109 INFO main lib_bscs:953 - Le PRM_DESCRIPTION correspondant aux co_id 28724,sncode 211
et le parameter_id 34 en paramètre est = Forfait plafonné 130 (crédit 71 DH)
20:39:37,125 INFO main lib_bscs:974 - Le deriner status et la valeur de Valid_From_Date correspondants
au co_id et shdes en paramètre n'existent pas
20:39:37,203 INFO main lib_bscs:1002 - la promotion est assignée pour ce client 1
```

Figure 13: résultat obtenu avec Eclipse.

Pour vérifier que le résultat obtenu avec Eclipse est correct, des tests de ces méthodes ont été effectués sous Toad For Oracle.

- La méthode qui retourne le SNCODE d'un service à partir de son SHDES

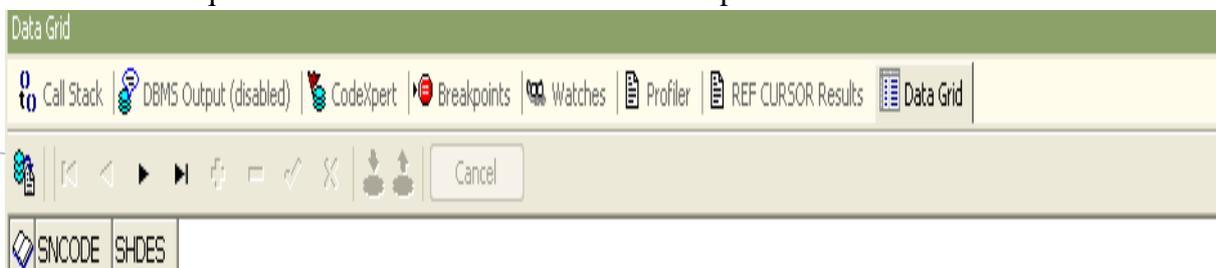
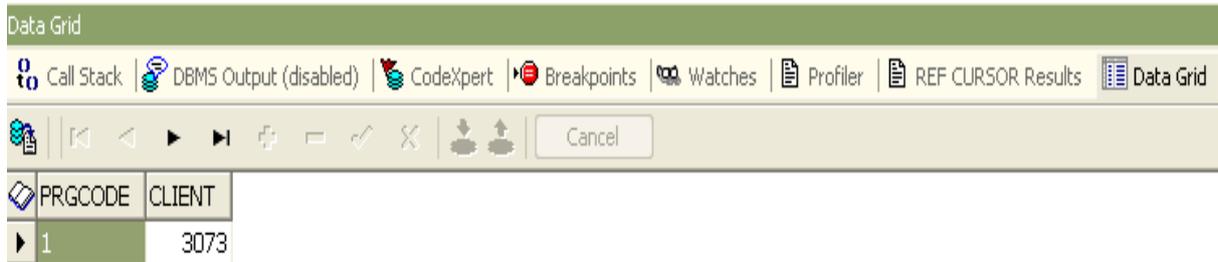


Figure 14: résultat obtenu avec Oracle de la méthode GetSncode(2)

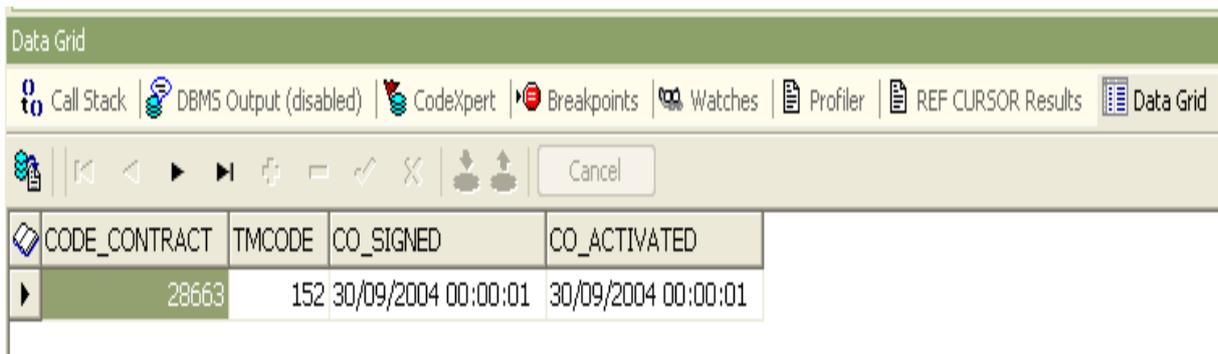
- La méthode qui retourne le code de la catégorie d'un client à partir de son numéro.



PRGCODE	CLIENT
1	3073

Figure 15: résultat obtenu avec Oracle de la méthode GetCatCustomer(5)

- La méthode qui retourne les informations sur un contrat donné.



CODE_CONTRACT	TMCODE	CO_SIGNED	CO_ACTIVATED
28663	152	30/09/2004 00:00:01	30/09/2004 00:00:01

Figure 16: résultat obtenu avec Oracle de la méthode GetContractInfo(6)

- La méthode qui retourne le dernier PT d'un contrat et le numéro du client avant une date donnée.



TMCODE	CUSTOMER_ID	CO_ID	TMCODE_DATE
102	4473	28513	01/08/2004

Figure 17: résultat obtenu avec Oracle de la méthode GetLastContractRp(7)

- La méthode qui retourne les informations de l'OVW d'un service pour un contrat donné.

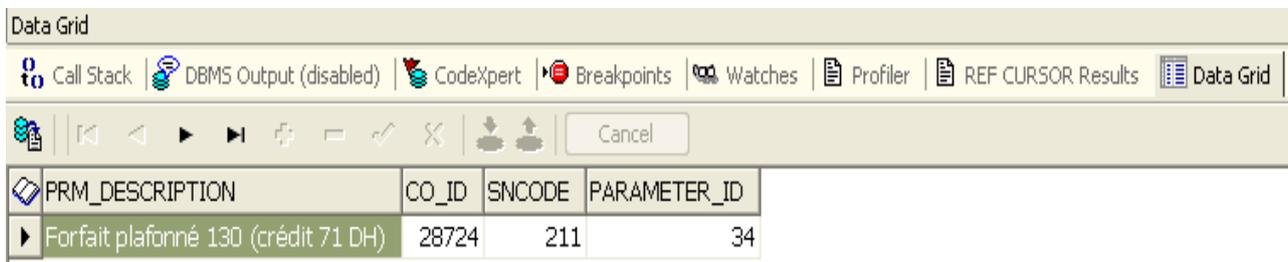




OVW_ADV_CHARGE	ADV_CHARGE	ADV_CHARGE_PRD
N	0.996765	1

Figure 18: résultat obtenu avec Oracle de la méthode GetOverWriteInfo(17)

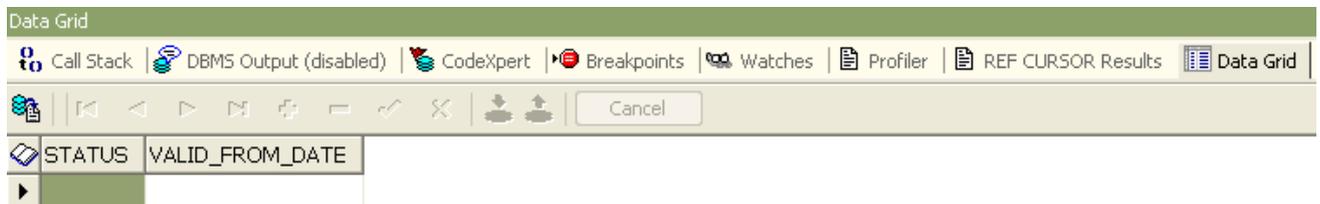
- La méthode qui retourne le dernier paramètre d'un service d'un contrat donné.



PRM_DESCRIPTION	CO_ID	SNCODE	PARAMETER_ID
Forfait plafonné 130 (crédit 71 DH)	28724	211	34

Figure 19 : résultat obtenu avec Oracle de la méthode GetSrvParam(8)

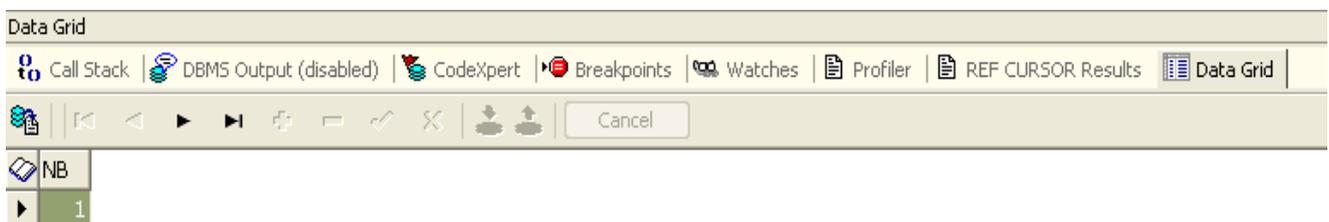
- La méthode qui retourne le dernier statut d'un service la date de sa validation.



STATUS	VALID_FROM_DATE

Figure 20: résultat obtenu avec Oracle de la méthode GetLastSrvStatus (11)

- La méthode qui vérifie si une promotion est assignée ou non pour un client donné.



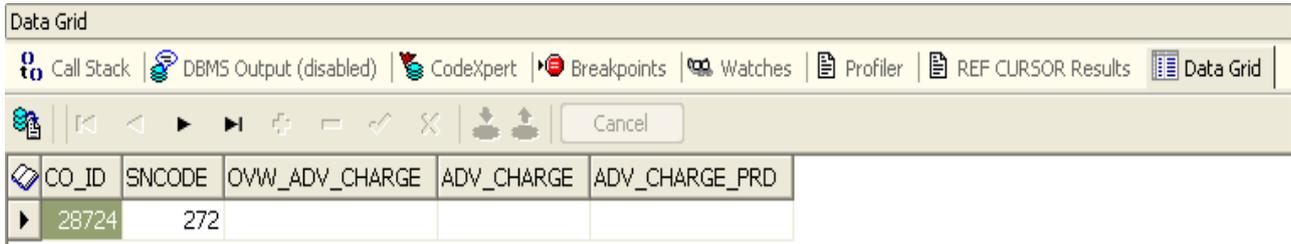
NB
1

Figure 21: résultat obtenu avec Oracle de la méthode CheckPromoAssigned (14)

La vérification de la méthode qui applique un OVW sur un service pour un contrat donné se fait au niveau de la BD.



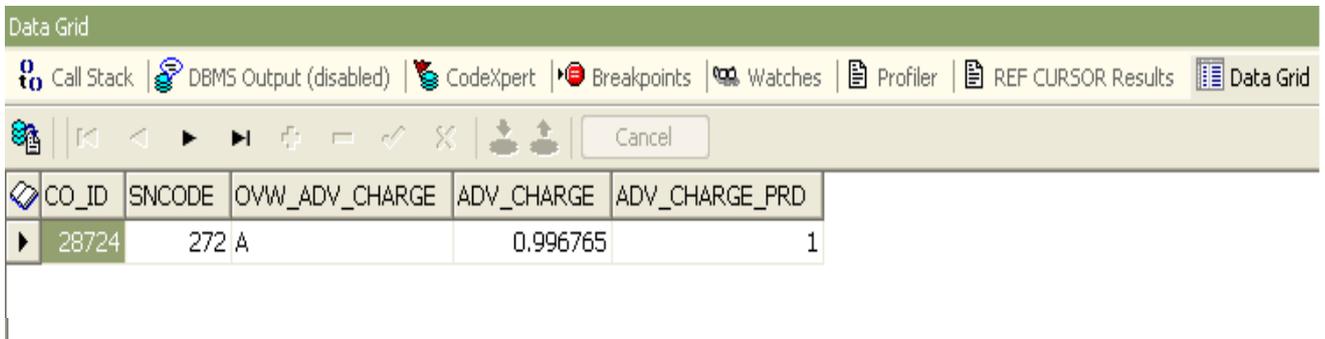
- Prise d'écran avant l'attribution d'OVW pour le contrat 28724.



CO_ID	SNCODE	OVW_ADV_CHARGE	ADV_CHARGE	ADV_CHARGE_PRD
28724	272			

Figure 22 : résultat obtenu avec Oracle de la méthode SetOverWriteSrv(16) avant l'application de l'OVW

- Prise d'écran après l'attribution d'OVW pour le même contrat.



CO_ID	SNCODE	OVW_ADV_CHARGE	ADV_CHARGE	ADV_CHARGE_PRD
28724	272 A		0.996765	1

Figure 23 : résultat obtenu avec Oracle de la méthode SetOverWriteSrv(16) après l'application de l'OVW

Conclusion :

Cette API restera valable même si MT élargit son réseau en ajoutant des nouveaux clients, des nouveaux services, des nouveaux contrats.....

Conclusion

Mon projet de fin d'études, effectué au sein de la société Atos Origin, a consisté à créer une API pour la factorisation et l'industrialisation des promotions fixes de Maroc Telecom. Une solution qui facilite le travail du programmeur des promotions.





Au cours de cette première période de stage, j'ai pu réaliser la partie factorisation du projet, Ma mission dans ce cadre a été déclinée en trois grandes phases. Dans une première, une étude a été faite sur des exemples des promotions fixes récentes de Maroc Telecom afin de déduire le traitement répétitif. La seconde phase a consisté en une étude sur les vingt cinq tables de la base de données de Maroc Telecom ayant pour but d'inférer le rôle de chaque table ainsi que la relation entre elles. Et enfin, la troisième et dernière phase a été consacrée à la réalisation de l'application.

La réalisation de ce projet a été une excellente mise en pratique de mes connaissances informatiques acquises tout au long de mon cursus surtout en bases de données. Ce stage m'a permis de compléter et d'approfondir ma formation dans le domaine d'informatique grâce à sa richesse technique et aux nombreuses notions utilisées.

Je peux donc conclure, que j'ai vécu une expérience professionnelle qui me sera utile durant ma future vie professionnelle.

Perspective

En perspective directe, je vais réaliser la deuxième partie du projet, qui consiste à l'industrialisation des promotions, puisque mon stage va continuer jusqu'au 15 aout.





Il s'agit de faire un modèle de données commun pour toutes les promotions à implémenter dans l'avenir, pour cela il faut étudier des exemples des scripts de création des tables des promotions récentes et dégager les champs en commun, lesquels il faut prendre en considération dans ce modèle de données:

- Une partie configuration: Une table (ou plusieurs) où configurer la nouvelle promotion (code de la promotion, date début, date fin, PT éligibles, services éligibles, paramètres des services éligibles, nombre de mois de la promotion, nombre de mois de tolérance,).
- Une partie données: les données des clients sur lesquels une promotion doit être appliquée: co_id, date_activation, date_debut_promo, date_fin_promo, montant_affecte(OCC),).

Dans ce modèle de données, il faut différencier entre les promotions standards et les promotions non standards. L'objectif étant que tous les développeurs de toutes les nouvelles promotions doivent se baser sur ce modèle de données pour adapter le jar créé dans la partie factorisation selon leur besoin.

Bibliographie et Webgraphie

[1]: DOCUMENT INTERNE : GROUPE ATOS ORIGIN

[2]: document interne « WIAM - PROJET REFONTE DU SI FIXE MAROC TELECOM », *Asmae Radouane*, 07/12/2004.

[3] : document interne DOSSIERS DE SPECIFICATIONS FONCTIONNELLES :

- ❖ « FC 4240 – PROMO 50% FIXE MARS 2011 », 27/12/2010

- ❖ « SFD_FC4240_BSCS-IF_Promo_50%_Fixe_Mars2011_V1.0.doc », 03/02/2011

[4] : document interne DOSSIER DE SPECIFICATIONS FONCTIONNELLES :

- ❖ « FSP4238 V1.0 Promotion Data et Internet-Mars2011 », 19/01/2011

- ❖ « FSP3984-Promo_DATA_NA_V1.0.doc», 14/10/2010





[5]: document interne:

- ❖ «formation SPRING : SRING BATCH TMA Maroc telecom», 7/02/2010
- ❖ «Etude SpringBatch expertise JEE STEEVE LOZANO», 03/09/2009
- ❖ «Srping par la pratique écrit par Arnaud Cogoluègnes»
- ❖ « Extrait des exigences Maroc Telecom concernant les Batches »
- ❖ <http://static.springsource.org/spring-batch/>
- ❖ http://fr.wikipedia.org/wiki/Spring_framework

[6] : [http://fr.wikipedia.org/wiki/Eclipse_\(logiciel\)](http://fr.wikipedia.org/wiki/Eclipse_(logiciel))

[7] : [http://fr.wikipedia.org/wiki/Toad_\(logiciel\)](http://fr.wikipedia.org/wiki/Toad_(logiciel))

- ❖ PDF : Optimisez vos développements et votre administration Oracle
- ❖ <http://toadfororacle.com/index.jspa>

[8] : [http://fr.wikipedia.org/wiki/Java_\(langage\)](http://fr.wikipedia.org/wiki/Java_(langage))

[9] : http://fr.wikipedia.org/wiki/Structured_Query_Language

- ❖ <http://wiki.01net.com/index.php/SQL>
- ❖ http://fr.wikipedia.org/wiki/Structured_Query_Language
- ❖ <http://litis.univ-lehavre.fr/~fournier/SGBD/coursPLSQLPrint.php>
- ❖ <http://fr.wikipedia.org/wiki/PL/SQ>
- ❖ <http://deptinfo.unice.fr/~grin/messupports/plsql.pdf>
- ❖ <http://www-poleia.lip6.fr/~lepape/ens/bd/cours/cours6.pdf>

