

Évolution d'une application de CRM au sein d'une SSII

Frédéric Chapelon

▶ **To cite this version:**

Frédéric Chapelon. Évolution d'une application de CRM au sein d'une SSII. Multimédia [cs.MM]. 2012. dumas-01063211

HAL Id: dumas-01063211

https://dumas.ccsd.cnrs.fr/dumas-01063211

Submitted on 11 Sep 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS
CNAM D'AUVERGNE

MEMOIRE

présenté en vue d'obtenir

le DIPLOME d'INGENIEUR CNAM

SPECIALITE : Informatique
OPTION : Réseaux, Systèmes et Multimédia

par

CHAPELON Frédéric

Evolution d'une application de CRM au sein d'une SSII

Soutenu le 13 Juin 2012

JURY

Président :	**K. BARKAOUI**
Membres :	**C. DEVEAUX**
	P. LLAMAS
	M. MISSON
	M. SCHNEIDER
	F. TOUMANI

Remerciements

Je tiens à remercier tout d'abord mon maître de stage, M. Michel MISSON, ainsi que M. Michel SCHNEIDER pour leur disponibilité et l'aide qu'ils m'ont apporté tout au long de ce travail et au cours de la rédaction de ce mémoire.

Je remercie également la société ALFA Informatique du groupe CEGI et plus particulièrement :

- M. William LEDIEU, responsable du service gestion interne, pour sa confiance et l'autonomie qu'il a su me laisser tout au long de ce stage.
- Les membres du groupe de travail pour leur implication dans ce projet et le temps qu'ils ont su lui consacrer.
- Le personnel du service gestion interne pour le temps passé ensemble, leurs conseils et leur professionnalisme.

Enfin je tiens à remercier M. Christian DEVEAUX qui, malgré son départ de la société durant ce stage, a su se rendre disponible, à l'écoute et s'est investi dans ce projet qu'il m'avait confié.

Glossaire

.Net : « .Net fait partie intégrante de beaucoup d'applications fonctionnant sous Windows et fournit des fonctionnalités de base à ces applications pour leur fonctionnement. […] Pour les développeurs, le Framework .Net fourni un modèle de développement complet et cohérent pour la création d'applications qui offrent des expériences utilisateur visuellement superbes et permettent des communications transparentes et sécurisées ». (*Microsoft)*

Access : Microsoft Office Access est une application éditée par Microsoft. Il fait partie de la suite bureautique Microsoft Office. Il est composé de plusieurs programmes : le moteur de base de données Microsoft Jet, un éditeur graphique, une interface de type Query By Example (QBE) pour manipuler les bases de données, et le langage de programmation Visual Basic for Applications (VBA) pour créer des applications.

ASPX : Active Server Page Extended. Les pages internet, connues également sous le nom de formulaires, sont la partie centrale pour le développement d'applications internet. Ces formulaires peuvent être contenus dans des fichiers ayant l'extension ASPX. Ces fichiers contiennent aussi bien des balises statiques HTML que des contrôles serveurs et utilisateurs où les développeurs mettent les contenus statiques et dynamiques nécessaires au fonctionnement de la page internet. Les fichiers d'extension ASPX sont des pages internet créées pour utiliser des scripts et qui se basent sur de la programmation ASP.Net. Elles sont utilisées par les serveurs internet. Leur code est mis à disposition par un serveur ASP.Net, généralement Microsoft IIS, et ensuite interprété par le navigateur du client comme du code HTML. Les fichiers de scripts ASPX peuvent être rédigés au sein de Microsoft Visual Studio et être visualisés par les navigateurs internet tels qu'Internet Explorer, Firefox, Opera ou Chrome.

BPMN: Business Process Modeling and Notation. Il s'agit d'un langage symbolique permettant de représenter des processus métier complexes sous forme de schéma.

Cache : Il s'agit d'une zone de stockage locale temporaire des données. Le principe de la mise en cache est de stocker localement, soit en mémoire, soit dans un fichier, des données temporaires situées sur un serveur distant. Ce principe est généralement utilisé dans les applications Web. Il permet d'améliorer les performances des applications en diminuant les

interrogations serveur. Lors de la première interrogation du serveur distant, les données sont rapatriées localement dans le cache du poste client pour être utilisées lors des interrogations suivantes. Cependant une gestion du cache doit être effectuée afin de ne pas travailler sur des données obsolètes.

Champs : En SQL, représente généralement les valeurs contenues dans une colonne d'une table. Pour une application graphique, représente une zone de saisie.

Clés ou Clé primaire : En SQL représente une ou plusieurs colonnes d'une table dont les valeurs permettent d'identifier sans équivoque une ligne de cette table.

CRM : Customer RelationShip Management. Représente la gestion des relations avec les clients. Ce sigle peut représenter à la fois une méthode ou par extension une application informatique.

CSV : Comma-separated values. A l'origine Il s'agit d'un format informatique ouvert représentant des données tabulaires sous forme de « valeurs séparées par des virgules ». Cependant, aujourd'hui, le caractère de séparation peut être un point-virgule, une tabulation ou encore deux points. Un fichier CSV est un fichier texte, par opposition aux formats dit « binaires ». Chaque ligne correspond à une ligne d'un tableau et les colonnes d'une même ligne sont séparées par le caractère de séparation. Une ligne est une suite ordonnée de caractères terminée par un caractère de fin de ligne.

Délégué : Un délégué est l'équivalent .Net d'un pointeur de fonction. Son rôle est d'appeler une ou plusieurs méthodes qui peuvent varier selon le contexte. Ces méthodes doivent posséder la même signature que le délégué.

DLL : Dynamic Link Library ou en français Bibliothèque de liens dynamiques. Il s'agit d'un ensemble de méthodes ou fonctions mises à disposition des applications pour leur fonctionnement.

Framework : Dans le cadre de Microsoft .Net, il s'agit de l'ensemble du système permettant de faire fonctionner une application .Net. Dans une architecture n-tiers, il s'agirait du serveur d'objet.

GTP : Groupe de Travail Projet. Il s'agit d'un groupe de réflexion et d'analyse composé de personnes de divers horizons dont l'objectif est de mener à bien un projet.

GRC : Sigle français de la CRM.

HTML : Hypertext Markup Language. Langage balisé utilisé pour la création des pages internet. Ce langage est interprété par les navigateurs internet.

Hyper-v : Service de virtualisation de poste de Microsoft.

IHM : Interface Homme Machine. En développement désigne généralement la partie graphique d'une application où des données peuvent être saisies par l'utilisateur.

Intergiciel : traduction française de Middleware.

JavaScript : C'est un langage de programmation de scripts principalement utilisé pour les pages Web interactives. Il permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes. Ce code est directement écrit dans la page HTML.

Mainframe : Ordinateur de grande puissance qui peut avoir une mémoire de grande capacité ainsi que de nombreuses unités de stockage afin de traiter de nombreuses opérations simultanément. Historiquement, un mainframe est associé à un système centralisé plutôt qu'à un système distribué.

Middleware : couche intermédiaire de logiciel permettant à deux ou plusieurs applications réparties dans un réseau de communiquer entre elles.

OSI : Open Systems Interconnection. Modèle de communications entre ordinateurs proposé par l'ISO (Organisation internationale de normalisation). Il décrit les fonctionnalités nécessaires à la communication et l'organisation de ces fonctions.

PERT : Program of Evaluation and Review Technique. Technique de gestion de projet qui permet de visualiser la dépendance des tâches et de procéder à leur ordonnancement. C'est un outil de planification. Pour chaque tâche, on indique une date de début et de fin au plus tôt et au plus tard. Le diagramme permet de déterminer le chemin critique qui conditionne la durée minimale du projet. Le but est de trouver la meilleure organisation possible pour qu'un projet soit terminé dans les meilleurs délais, et d'identifier les tâches critiques, c'est-à-

dire les tâches qui ne doivent souffrir d'aucun retard sous peine de retarder l'ensemble du projet.

Réplication : processus de partage d'informations pour assurer la cohérence de données entre plusieures sources de données redondantes, pour améliorer la fiabilité, la tolérance aux pannes, ou l'accessibilité.

Script : Liste de commandes écrites dans un certain langage informatique destinée à être interprétée pour effectuer une certaine tâche. Pour fonctionner, les scripts doivent être interprétés par un programme ou un serveur dédié au langage dans lequel ils ont été écrits.

SDK : Software Development Kit. Ensemble d'outils permettant aux développeurs de créer des applications de type défini. Ils sont en général fournis avec les langages de programmation propriétaires ou certaines applications spécifiques.

Service Web : programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, ne nécessitant pas d'intervention humaine.

SGBD : Système de Gestion de Base de Données. C'est un outil informatique qui permet d'insérer, de modifier et de rechercher efficacement des données spécifiques dans une grande masse d'informations partagées entre tous les utilisateurs. Il comprend un ensemble de données liées (la Base De Données), et un ensemble de programmes permettant de manipuler et d'accéder à ces données. Il doit aussi assurer le contrôle, la sécurité et la confidentialité des données. Différentes solutions existent sur le marché, les plus importantes étant Oracle et Microsoft SQL server.

Snapshot : Capture instantanée. C'est la photographie, à un moment précis, de l'état d'un système informatique.

SQL : Structured Query Language. Langage informatique destiné à interroger ou piloter une base de données.

TSE : Terminal Server Edition. Service Windows permettant de partager, avec plusieurs utilisateurs, l'utilisation d'un serveur et de ses applications au moyen de clients légers. La

charge de traitement est alors effectuée par le serveur et le client léger ne sert que d'interface graphique. Son équivalent dans le monde Linux pourrait être considéré comme les Terminaux X.

UML : Unified Modeling Language. Langage graphique d'analyse et de conception orienté objet pour la modélisation des données et des traitements. C'est une formalisation très aboutie et non-propriétaire de la modélisation objet utilisée en génie logiciel.

VB6 : Visual Basic version 6. Environnement de développement permettant de créer des applications utilisant le langage Visual Basic de Microsoft.

Virtualisation : Il s'agit de l'ensemble des techniques permettant de faire fonctionner simultanément sur une seule machine physique plusieurs systèmes d'exploitation. Ces systèmes sont séparés les uns des autres, comme s'ils fonctionnaient sur des machines physiques distinctes. Les principaux acteurs sur le marché de la virtualisation de postes sont VMWARE et Microsoft avec sa solution Hyper-v.

Workflow : flux de travaux représentant les opérations à réaliser par une personne, un groupe de personnes, un organisme pour accomplir l'ensemble des tâches ou activités regroupées en un même processus métier. Le workflow permet la modélisation des processus métier dans le cadre d'une démarche plus globale BPM.

Chapitre 1 Contexte 4
1.1 Présentation de l'entreprise 4
1.2 Présentation du sujet 6
1.3 Présentation du logiciel *Selligent* 7
1.3.1 Présentation générale du logiciel 7
1.3.2 Rôle et évolution du logiciel au sein de l'entreprise 8
1.4 Présentation du logiciel *Microsoft CRM* 9
1.4.1 Présentation générale du logiciel 9
1.4.2 Rôle et évolution du logiciel au sein de l'entreprise 11
1.5 Agencement des applications au sein du système d'information 12
Chapitre 2 Méthodologie 15
2.1 Résultat du découpage global 15
2.1.1 Description des étapes 15
2.1.2 Agencement des étapes 20
2.2 Analyse 21
2.2.1 But de l'analyse 21
2.2.2 Choix de la méthode et des outils 21
2.2.3 Déroulement de l'analyse 22
2.3 Récupération de données 23
2.3.1 Considérations sur la récupération de données 23
2.3.2 Méthodes d'importation 24
2.3.3 Choix de la méthode de récupération de données 26
Chapitre 3 Travail réalisé 28
3.1 Installation d'une plateforme de test 28
3.2 Migration SQL 2008 sur la plateforme de test 29
3.3 Migration CRM 4.0 sur la plateforme de test 30

3.3.1 Migration du logiciel 30
3.3.2 Modification des personnalisations 31
3.4 Analyse des processus commerciaux 35
3.5 Création et personnalisation des écrans sur la plateforme de test 36
3.6 Reprise de données sur la plateforme de test 36
3.7 Ecart entre le planning prévisionnel et réalisé 42
Chapitre 4 Retour d'expérience 45
Chapitre 5 Prolongement du projet 48
Annexe 1 Business Process Modeling and Notation (BPMN) 52
Annexe 2 Compte-rendu de réunion GTP CRM Commercial 59
Annexe 3 Concepts commerciaux 74
Annexe 4 Les différentes architectures d'applications 77

Introduction

La concurrence sur les différents marchés, la volatilité des clients des entreprises, le développement des nouveaux moyens de communication et d'information numérique ont conduit à un changement de relation entre les entreprises et leurs clients. Ces dernières, pour être plus pérennes, ont dû s'adapter et rechercher les meilleurs moyens de fidéliser leur clientèle. La CRM (Customer RelationShip Management) ou GRC (Gestion des Relations avec les Clients) représente à la fois un concept et une gamme d'applications visant à intégrer ces changements au sein des entreprises.

Sous sa forme de concept la CRM consiste à recentrer son activité autour du client afin de créer une relation durable avec ce dernier. Pour atteindre ce but, il faut avoir une vision détaillée du client, de ses habitudes d'achat, de ses besoins afin d'établir un climat de confiance et de lui proposer des solutions personnalisées adaptées. La communication des informations sur le client, obtenues par les différents services en relation avec lui, doit permettre de mieux l'appréhender et de lui donner vraiment l'impression d'être connu et non pas d'être un client parmi d'autres. L'entreprise tire bénéfice de cet échange en identifiant ainsi les bons clients, ce qui lui permet d'axer son action sur leur contentement et donc de travailler à long terme avec eux. La mise en place de ce concept implique une méthodologie, des processus et une culture d'entreprise orientés vers le client ; mais aussi des outils permettant de gérer et d'exploiter les informations recueillies.

La CRM applicative sert de support à ces concepts. Elle peut être classée en trois sous-systèmes : la CRM opérationnelle, l'analytique et la collaborative. La CRM opérationnelle se résume à l'intégration et l'automatisation des processus horizontaux correspondant à tous les contacts directs avec le client quels que soient les canaux de communication. Ce sous-système inclut notamment les processus de vente, de marketing et de support client.

La CRM analytique permet, au travers de l'analyse des informations collectées, de dégager une vue d'ensemble des besoins et des attentes des clients. Ce sous-système permet de mieux comprendre les comportements des clients au travers de l'analyse de leurs habitudes, des profits ou des produits.

La CRM collaborative sert de support de collaboration et de communication avec les clients, les partenaires et les fournisseurs. Elle regroupe tous les canaux d'échange avec le client et avec les partenaires à propos du client. Ces échanges sont essentiellement destinés à améliorer la rentabilité et la fidélité du client.

Ce mémoire a pour objet de présenter le travail effectué dans le cadre d'un changement de logiciel de CRM opérationnelle, ce changement s'intégrant dans un projet plus vaste de regroupement des différents sous-systèmes CRM au sein d'une même solution applicative. Ce projet a été conduit au bénéfice d'une société d'édition de logiciels de gestion pour le domaine associatif. Pour développer notre sujet d'analyse, nous verrons dans un premier chapitre le contexte du changement, puis la méthodologie employée pour mener à bien ce projet. Le chapitre trois présentera quant à lui le travail réalisé dans le cadre de la préparation de ce mémoire d'ingénieur. Les chapitres quatre et cinq présenteront respectivement l'expérience que j'ai pu retirer de ce projet et le prolongement de ce dernier à la fin de ce stage.

Chapitre 1 Contexte

1.1 Présentation de l'entreprise

La société ALFA Informatique (Auvergne Logiciel Formation et Assistance Informatique) a été fondée en 1980 par Claire DEVEAUX et Jean-François BAZIN, rejoints en 1982 par Christian DEVEAUX. Elle conçoit et réalise des progiciels de gestion pour les secteurs sanitaire, médico-social et social. Son siège social se situe à Clermont-Ferrand dans la zone industrielle du Brézet.

Alfa informatique dispose de plusieures agences commerciales situées dans les grandes villes Françaises (Strasbourg, Lyon, Toulouse, Rennes, Lille et Paris). Cette disposition lui assure une couverture quasi intégrale du marché français, lui autorisant ainsi une certaine réactivité aux demandes des clients.

En 2007, la société a été rachetée par le groupe CEGI, principal concurrent sur son marché, dont le siège social est à Paris. Le rapprochement de ces deux sociétés a permis de créer un groupe leader sur le marché national. Ce groupe compte aujourd'hui environ 500 salariés dont 170 pour ALFA informatique. La société ALFA Informatique a conservé son statut de SAS (Société par Actions Simplifiée) au capital d'un million d'euros. Monsieur Jean-François BAZIN est devenu Directeur général d'ALFA Informatique pendant que Monsieur Gérard TAIEB, fondateur du groupe CEGI, en devenait Président directeur général. En 2010, la société dégageait un chiffre d'affaire de 13,5 millions d'euros, en très léger recul par rapport à l'année précédente.

La société ALFA Informatique accompagne, forme et crée des logiciels pour les associations de type loi 1901 et les entreprises qui interviennent dans des secteurs tel que: la pauvreté et l'exclusion, l'aide aux personnes âgées, l'aide aux handicapés, le domaine hospitalier, l'enfance et la famille ou la politique de santé.

Ainsi on retrouve par exemple des associations comme l'Armée du salut, des maisons de retraite ou encore des organismes de gestion d'associations.

Ce secteur d'activité ayant des contraintes particulières, ALFA Informatique va se servir de son savoir-faire acquis dans le domaine pour réaliser des produits qui vont correspondre aux besoins de ses clients. Ces produits sont regroupés en 4 familles :

- alfaFinance propose des logiciels pour la comptabilité. Ceux-ci contiennent toutes les règles comptables, financières et budgétaires spécifiques au secteur des clients.
- alfaGrh est une gamme de produits qui facilitent la gestion des ressources humaines au sein de la structure organisationnelle du client.
- alfaSanté regroupe des progiciels spécifiques à la gestion des structures de soins.
- alfaSocial fournit des produits adaptés aux spécificités des différentes filières de prise en charge sociale et médico-sociale.

L'entreprise est organisée en sept services : le service administratif, le service client, le service commercial, le service développement, le service gestion interne, le service formation et le service offre client.

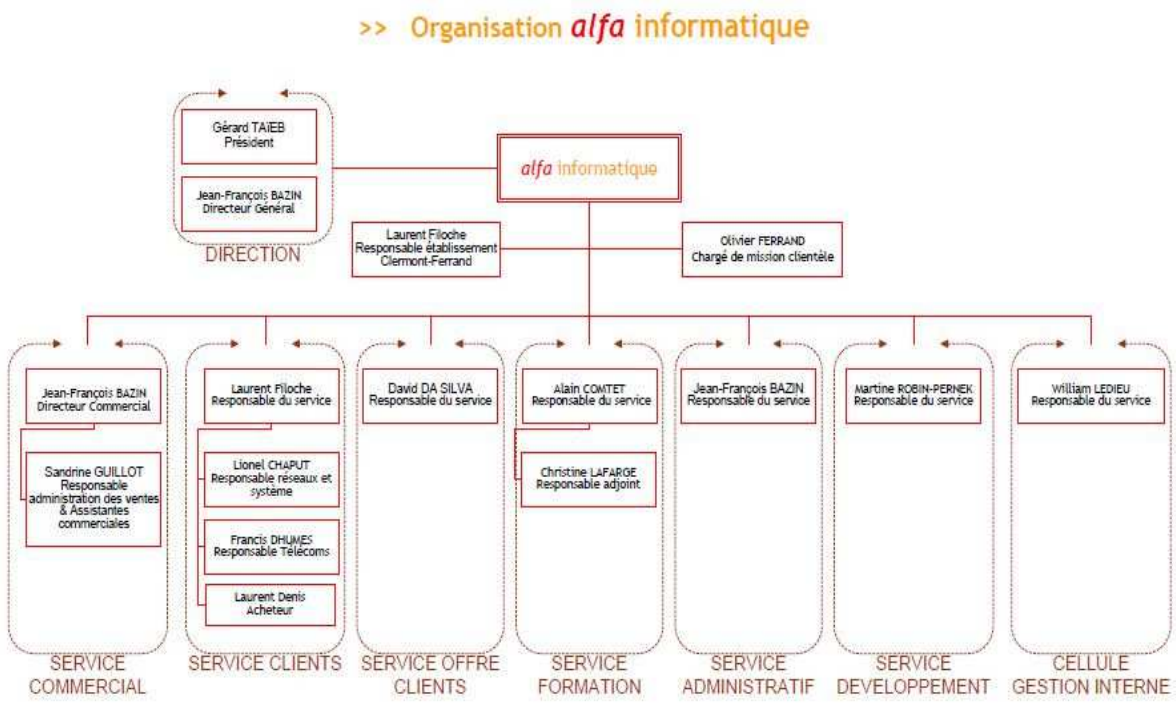


Figure 1 : Organigramme général de la société ALFA Informatique

J'ai effectué mon stage au sein du service gestion interne, sous la responsabilité de Monsieur William LEDIEU. Ce service a la responsabilité de toute l'informatique interne de l'entreprise, aussi bien matérielle que logicielle.

Pour ce faire, il se décompose en deux pôles de compétences : le pôle développement et le pôle technique. Le pôle technique, composé de deux techniciens, a la responsabilité du maintien et de l'évolution de l'infrastructure réseau et système.

Le pôle développement, au sein duquel j'ai effectué mon stage, est composé de deux chefs de projet. Il a la responsabilité du maintien, de l'évolution et de la création des applications utilisées par les salariés de l'entreprise.

1.2 Présentation du sujet

La société ALFA Informatique utilise depuis 1996 le logiciel de gestion commerciale *Selligent* développé par la société du même nom. Malgré les nombreuses adaptations apportées par le service informatique au cours de ces dernières années, certaines limitations sont apparues au fil du temps, conduisant aujourd'hui à une réflexion visant à changer de solution logicielle pour faciliter le suivi des clients.

La solution *Microsoft CRM* est déjà utilisée pour le traitement des incidents des clients. Afin de centraliser l'ensemble des informations liées à ces derniers dans un unique logiciel CRM, une réflexion antérieure à ce mémoire, a conduit à retenir cette solution pour intégrer l'ensemble des données commerciales existantes.

La reprise des données du logiciel CRM commercial existant doit être l'occasion d'analyser et de réétudier l'ensemble des processus commerciaux déjà en place. Par ailleurs, en 2007, la société ALFA informatique est devenue une filiale de CEGI. L'ensemble du groupe utilisant le même logiciel commercial, les besoins des filiales de ce dernier doivent être pris en compte.

Ce stage, débuté en février 2010, avait donc pour but d'implémenter une nouvelle solution logicielle commerciale pour mieux répondre aux besoins de suivi des clients du groupe en s'appuyant sur l'offre CRM proposée par Microsoft. Cette étude devait prendre en considération les données existantes (logiciel *Selligent*), les besoins actuels, notamment en termes de mise à disposition d'informations et d'échange de données avec des applications périphériques (CRM Hotline, base clients), mais aussi les besoins futurs (mise à disposition d'informations pour les clients via un site Extranet).

Pour mieux appréhender le travail effectué, nous allons tout d'abord nous pencher sur les deux logiciels concernés par ce projet.

1.3 Présentation du logiciel *Selligent*

1.3.1 Présentation générale du logiciel

Le logiciel *Selligent*, dans sa version actuellement utilisée au sein de la société ALFA Informatique, est une application d'architecture deux-tiers [**Voir annexe 4**] s'appuyant sur un serveur de base de données de type Microsoft SQL (Structured Query Language). Il s'agit d'un logiciel de CRM permettant de gérer les clients au niveau commercial au travers d'un portefeuille clients, en allant du devis à la facturation en passant par les campagnes commerciales. Il permet également la prise en charge d'une base de connaissances et de mécanismes de création et de résolution d'incidents clients.

Le principe de cette application est de fournir un environnement évolutif au sein duquel toutes les informations relatives à un client peuvent-être traitées. Cette solution propose donc un squelette applicatif contenant des mécanismes et des notions commerciales de base. Ce squelette sera alimenté et personnalisé par la suite afin d'être au plus proche des mécanismes commerciaux de l'entreprise dans laquelle il est implémenté.

Ces personnalisations se font au travers d'une interface graphique permettant de renseigner des champs de choix multiple, de rajouter des champs de saisie ou encore de créer des scripts dans un langage propriétaire. Ces derniers peuvent se déclencher sur un certain nombre d'évènements tels que l'ouverture, l'enregistrement ou la fermeture de fiches clients.

Cette solution offre également des possibilités de réplication partielle de bases de données permettant de scinder l'ensemble des données en portefeuilles commerciaux organisés selon certains critères. Cette faculté permet donc aux utilisateurs nomades de travailler en mode déconnecté sur une base partielle moyennant la mise en place d'un mécanisme de réplication.

1.3.2 Rôle et évolution du logiciel au sein de l'entreprise

Ce logiciel est utilisé depuis 1998 au sein de la société et de ce fait l'organisation de base a été amenée à évoluer au fil des versions de l'éditeur et des modifications des méthodes commerciales de l'entreprise. Certaines notions présentes dans la version actuelle n'ont pas été utilisées pour des raisons historiques et d'autres n'ont plus été utilisées du fait de leur inadéquation avec l'évolution des méthodes de travail.

Si l'on prend par exemple les mécanismes de création et de résolution d'incidents, ceux-ci n'étaient pas présents dans les premières versions du logiciel. Un logiciel interne avait donc été créé pour ce rôle. Il a été par la suite conservé puis remplacé dernièrement par la solution CRM de Microsoft.

Le système de base nomade mettant en jeu les mécanismes de réplication a fini par être abandonné à cause des nombreux problèmes rencontrés inhérents à la réplication tels que les conflits de réplication liés à la modification de mêmes données par deux utilisateurs, ou encore de temps de réplication excessifs liés à l'augmentation constante de la taille des bases de données. D'autre part, l'avènement de nouvelles technologies permettant une connexion quasi permanente au réseau d'entreprise a été prépondérant dans la décision d'abandonner le système de réplication au profit de solutions en mode connecté faisant appel à des serveurs Microsoft TSE (Terminal Server Edition).

Au fil du temps les mécanismes de devis ne correspondaient plus aux règles commerciales de la société et le logiciel ne pouvait pas être adapté pour être en adéquation avec elles. Les devis ont donc été externalisés au sein d'une application propre à l'entreprise. Ceux-ci étant liés aux produits, ces derniers ont également été externalisés dans une application tierce.

Aujourd'hui le nombre d'utilisateurs de l'application *Selligent* est de l'ordre d'une soixantaine de personnes au sein du groupe, l'essentiel de ces utilisateurs appartenant au service commercial. Du fait d'un système de licences d'accès au logiciel par utilisateur, la consultation du fichier clients au travers de *Selligent* n'était pas une solution envisageable économiquement. Une application tierce a donc été créée dont le rôle est de contenir et de donner accès en consultation au fichier clients à l'ensemble des collaborateurs du groupe.

Les évolutions des champs de saisie au fil du temps telles que le passage d'un champ texte à une zone de choix multiple, d'une zone libre à un champ obligatoire, la réutilisation d'un champ obsolète pour stocker de nouvelles données ont conduit à une perte de la cohérence de la base de données. Le choix a donc été fait d'effectuer un travail de consolidation des données au sein de l'application gérant le fichier clients et de limiter l'accès en modification sur les données clients à un certain nombre d'utilisateurs. Il s'agissait de rendre cette application maîtresse des données de l'application *Selligent* en ce qui concerne les informations sur les clients.

Au cours des années, le rôle de l'application *Selligent* a donc été de plus en plus confiné à un rôle purement commercial se contentant de gérer des informations sur les opportunités, les actions et les campagnes commerciales [**Voir annexe 3**], délégant la gestion des autres informations à des applications tierces pour la plupart développées en interne. Le rôle de moins en moins central de cette application et son manque d'évolution on conduit à envisager son remplacement. La direction d'ALFA Informatique, le service commercial et le service client se sont donc concerté et on fait le choix de prendre la solution *Microsoft CRM*. Ce choix permettant d'avoir un seul est même logiciel au sein de l'entreprise pour la gestion des clients au niveau hotline et commercial.

1.4 Présentation du logiciel *Microsoft CRM*

1.4.1 Présentation générale du logiciel

Comme pour le logiciel *Selligent*, la solution de Microsoft est, comme son nom l'indique, un logiciel de CRM. Nous allons donc retrouver les mêmes fonctionnalités, à savoir : la gestion d'un portefeuille clients et de tous les services et besoins relatifs aux clients ainsi que la mise à disposition d'un squelette applicatif évolutif.

La différence fondamentale va se retrouver dans la technologie employée. En effet la solution *Microsoft CRM* est une application Web d'architecture n-tiers [**Voir annexe 4**]. Étant donné qu'il s'agit de la dernière génération d'applications, elle va pouvoir bénéficier des nouvelles technologies. De plus, étant une solution Microsoft, elle s'intègre parfaitement avec les autres solutions Microsoft notamment avec les outils bureautiques. A l'instar de la solution *Selligent*, elle propose un système de fonctionnement en mode déconnecté. Pour

cela elle utilise le système de réplication du logiciel de messagerie Microsoft Outlook grâce à un interfaçage avec ce dernier.

Son architecture n-tiers (voir Figure 2) s'articule autour d'un serveur Web fournissant des pages utilisant la technologie .Net de Microsoft [1], celui-ci ayant le rôle de serveur d'application. Ces pages vont fournir au client une interface au travers de son navigateur internet. Elles sont composées de code HTML (Hypertext Markup Language), de scripts en langage JavaScript et de contrôles serveur. Les deux premiers vont être responsables de l'interface homme machine et du contrôle de saisie, les contrôles serveurs vont quant à eux être les déclencheurs d'actions auprès du serveur d'application.

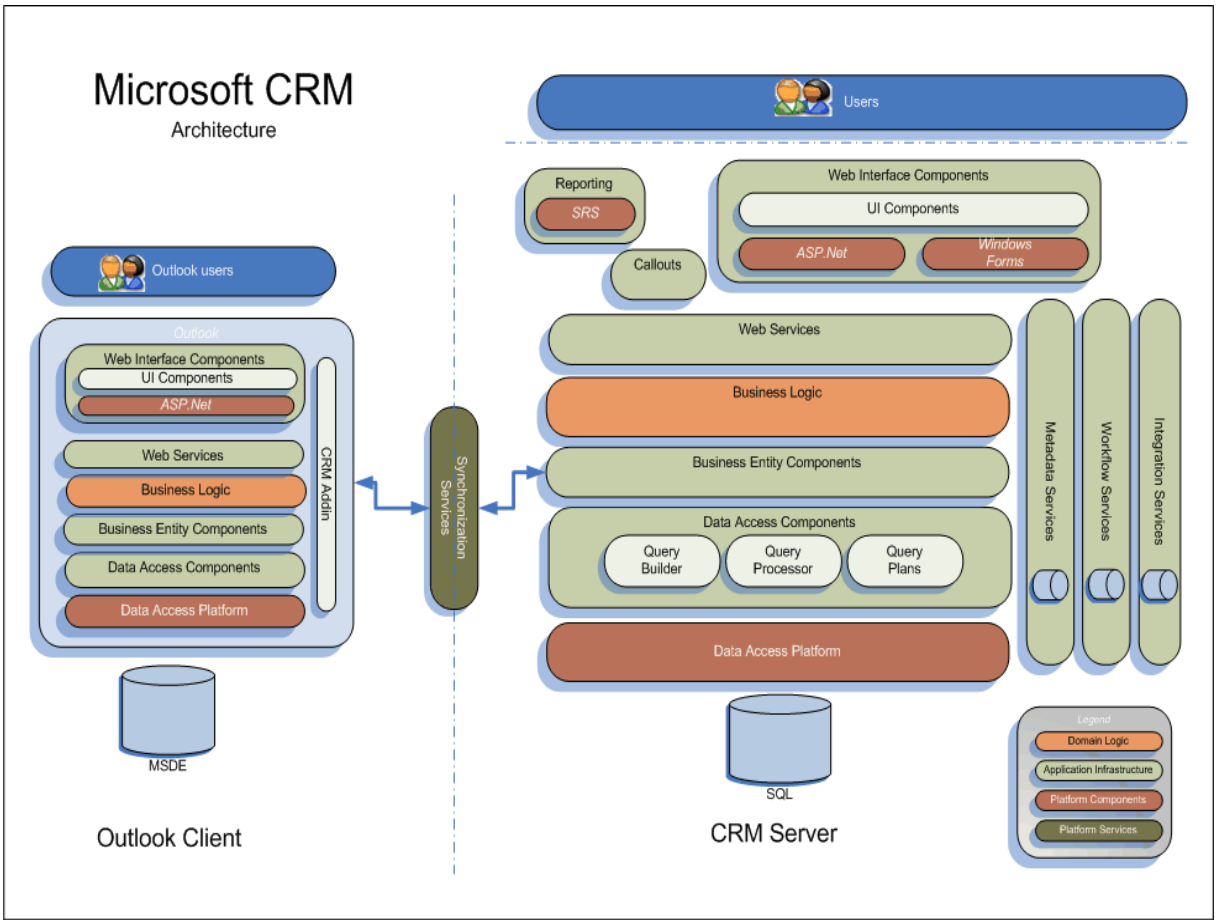


Figure 2 : Schéma de l'architecture de *Microsoft CRM* [2]

Le serveur d'application va avoir pour rôle de gérer la logique de traitement des pages et la mise à disposition de méthodes d'accès aux objets de l'application au travers de services Web. Pour cela, il va utiliser des objets mis à disposition par le serveur d'objet.

Le serveur d'objet s'appuie sur le Framework Microsoft. Son rôle est de mettre à disposition les objets de l'application et de gérer leur permanence. Les objets mis à disposition sont divers et variés, ils vont de la simple description d'objet à la gestion de la sécurité en passant par l'accès aux données.

Les données sont quant à elles stockées sur un serveur de base de données SQL et peuvent être verrouillées pour n'être accessibles qu'au travers des objets de l'application.

Parallèlement à cela, des services permettent d'accéder directement à la structure de la base de données ou encore de gérer des flux de travail. Des briques supplémentaires peuvent-être également rajoutées pour un interfaçage avec des applications tierces telles que l'application de gestion de rapport Microsoft Reporting Services.

1.4.2 Rôle et évolution du logiciel au sein de l'entreprise

Au sein de la société, ce logiciel est utilisé depuis 2003 à des fins de hotline technique en remplacement d'un logiciel propriétaire de gestion des appels. Cette utilisation restreinte de l'application, en comparaison de ses possibilités, s'explique par la présence d'applications tierces prenant en charge les autres fonctionnalités, comme par exemple *Selligent* pour la partie commerciale.

Bien que l'utilisation de la partie de gestion des appels nécessite la présence dans le logiciel des informations sur les clients, la solution CRM de Microsoft n'est pas l'application gérant le fichier clients. Une application tierce ayant ce rôle alimente et met à jour la base de données du logiciel de CRM.

Dans son utilisation, le logiciel de CRM sert non seulement à la gestion de la résolution des appels mais également à la consultation de ces appels par les collaborateurs du groupe CEGI étant en relation avec les clients ; ce qui représente aujourd'hui environ 250 utilisateurs. Ce chiffre est amené à être étendu à terme à l'ensemble des collaborateurs du groupe.

Depuis sa mise en service de nombreuses modifications ont été apportées tant par les mises à jour de versions de l'éditeur que par des développements internes à l'entreprise. Ces derniers ont pour but de personnaliser le squelette applicatif afin qu'il corresponde aux besoins du service chargé de la Hotline.

On peut distinguer deux grandes familles de développement interne :

- Les adaptations effectuées à l'aide de l'interface du logiciel de CRM. Celles-ci permettent de gérer la structure des données en créant des notions d'entité, que l'on peut assimiler à un objet représentant une vue logique des données dans la base de données et dont les attributs correspondent aux champs des tables. Elles donnent également la possibilité de créer les relations entre ces entités. Enfin, elles permettent de gérer le code JavaScript exécuté sur le poste client.
- Les adaptations effectuées au travers de développements de pages intégrées dans le logiciel de CRM. Il s'agit de développer des pages utilisant la technologie .Net de Microsoft et s'appuyant sur le modèle d'objet du logiciel de CRM fourni au travers des services Web de l'application et d'un SDK (Software Development Kit). Ces pages sont ensuite intégrées dans le logiciel de CRM au travers d'un fichier de configuration.

Selligent et *Microsoft CRM* sont donc deux briques faisant partie de l'ensemble des applications permettant la gestion des clients au sein de la société. Nous allons voir maintenant comment elles s'imbriquent pour former le système d'information.

1.5 Agencement des applications au sein du système d'information

Le système d'information s'articule aujourd'hui autour d'une gamme complexe de logiciels, pour la plupart développés en interne, ayant des interactions plus ou moins fortes entre eux.

Le point central de ce système est le fichier clients géré par le logiciel Alfa.tiers développé en interne. Les fonctions de ce logiciel sont de consolider les informations clients éparpillées dans les diverses bases de données des autres applications. Il sert également de point d'entrée et de mise à jour de ces informations à destination de ces autres applications. Il va mettre notamment à jour les informations clients utilisées dans les logiciels *Selligent* et *Microsoft CRM*.

Microsoft CRM, quant à lui, est utilisé pour sa partie gestion des appels ou, de manière plus générale, du suivi des demandes clients, de leur saisie jusqu'à leur résolution.

Selligent, comme nous l'avons vu précédemment, est utilisé pour sa partie purement commerciale. Il se limite donc à la gestion des opportunités, des actions et des campagnes [**Voir annexe C**] et délègue la gestion des devis et des articles à des logiciels tiers développés en interne.

La gestion des articles commercialisés et l'édition du catalogue est orchestrée par le logiciel Alfa.article.

Ces informations sont par la suite utilisées par le logiciel Alfa.devis dont le rôle est de gérer les devis et leurs réactualisations. Les devis sont générés à partir de modèles et ils sont ensuite insérés dans le logiciel commercial *Selligent*.

Lors des commandes clients, ces devis sont transformés au travers d'Alfa.devis pour être intégrés au logiciel de gestion commerciale Alfa.gesco puis au logiciel de comptabilité CEGI First.

Les contrats clients sont quant à eux gérés par l'application Alfa.maintenance tandis que le planning des interventions liées aux incidents ou aux commandes clients est organisé à partir du logiciel Alfa.Intervention.

Enfin, une interface Web mise à disposition du client lui permet de suivre l'évolution de ses incidents hotline au travers de l'application Alfa.site Web.

D'autres applications internes gravitent autour de ces logiciels. Il s'agit pour l'essentiel d'applications de gestion de rapports permettant notamment d'élaborer des rapports tels que des tableaux de bord pour les ingénieurs commerciaux (Alfa.ca). D'autres applications permettent d'obtenir le chiffre d'affaires correspondant aux factures à établir (alfa.fae), un suivi des actions engagées auprès des clients (alfa.suivi client) ou encore différentes vues concernant les clients et les ingénieurs commerciaux (alfa.suivi financier).

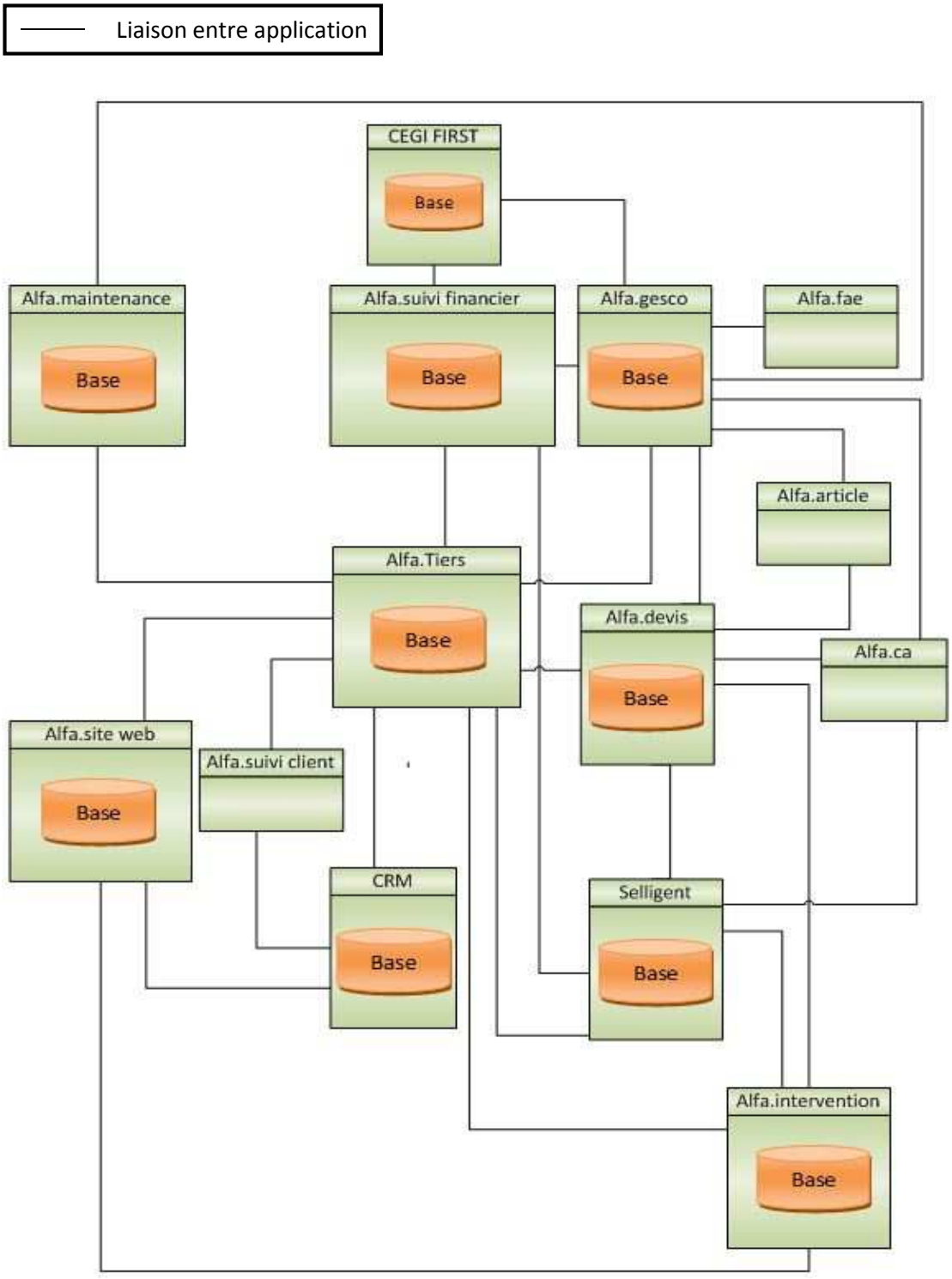


Figure 3 : Cartographie simplifiée des liens entre applications du système d'information

L'ensemble de ces applications forme donc un microcosme dense, complexe de par la variété des technologies utilisés (Access, VB6, .Net) et fragile du fait des interactions entre les diverses applications.

Le contexte dans lequel j'ai effectué mon stage étant présenté, nous allons maintenant voir la méthodologie employée pour mener à bien ce projet.

Chapitre 2 Méthodologie

Afin d'atteindre notre objectif, j'ai élaboré en début de projet un découpage global me permettant de mettre au point un planning agençant les tâches que j'avais identifiées. Par la suite, j'ai recherché une méthode pour mener à bien l'analyse de ce projet ainsi que des outils pour en retranscrire le résultat. Enfin, j'ai recherché par quel biais les données de l'ancien système pouvaient être récupérées et incorporées dans la nouvelle solution.

2.1 Résultat du découpage global

Le but de ce découpage était d'identifier les différentes étapes inhérentes à la réalisation de ce projet.

Lors de ce travail il a été constaté, avec ma direction, que des évolutions technologiques avaient eu lieu entre l'élaboration du sujet de stage et le début effectif du projet. Il a donc été décidé de les intégrer au stage.

Le choix a donc été fait de profiter du projet pour basculer le moteur de base de données de la version Microsoft SQL 2005 vers la version Microsoft SQL 2008 et de changer la version du CRM Microsoft en la passant de la version 3 à la version 4.

Ces étapes nécessitant d'être testées, elles ne pouvaient être effectuées sur l'environnement d'exploitation. Il s'est donc révélé nécessaire que je mettre en place une plateforme de test permettant d'effectuer tous les essais dans un environnement similaire à celui de production.

Dans un premier temps, j'ai identifié les étapes du projet et j'en ai estimé un temps homme afin de pouvoir par la suite les agencer dans le temps et obtenir une feuille de route du projet dont voici le résultat.

2.1.1 Description des étapes

Chaque étape a été identifiée en utilisant les caractéristiques suivantes : une lettre, une description, une estimation du temps nécessaire à sa réalisation et les personnes participant à sa réalisation. Etant la seule personne ressource technique affectée à ce projet, ces étapes étaient généralement réalisées par mes soins, sauf dans certains cas particuliers précisés dans le descriptif de l'étape. Les durées des étapes ont été calculées au plus court en se

basant sur les connaissances que j'avais en début de projet. Le projet global avait une date butoir définie à décembre 2011 dépassant largement le cadre de ce stage.

Etape A : Installation d'une plateforme de test

Durée estimée : 10 jours

Description :

Il s'agit de constituer une plateforme de test identique au système informatique de production tout en étant séparé de ce dernier. Cependant, les postes de cette plateforme doivent être accessibles par l'équipe du pôle développement à partir du réseau d'exploitation et les membres de l'équipe doivent pouvoir accéder aux ressources de leurs postes lorsqu'ils sont connectés sur les machines de la plateforme de test. Pour ce faire un serveur physique sera installé avec le système d'exploitation Windows serveur 2008 R2 Datacenter. Son rôle sera d'héberger, grâce à la technologie Microsoft Hyper-v, les machines virtuelles nécessaires au fonctionnement de la plateforme de test et de développement.

Etape B : Migration SQL 2008 sur la plateforme de test

Durée estimée : 1 jour

Réalisé par : pôle développement

Description :

Cette étape a pour but de tester la migration du moteur SQL 2005 de production vers la version SQL 2008 et de résoudre les problèmes rencontrés.

A cette occasion une procédure de migration du moteur SQL sera rédigée afin de pouvoir effectuer la migration sur le serveur d'exploitation. Les applications utilisant les bases de données stockées sur ce serveur devront être testées.

Etape C : Migration CRM 4.0 sur la plateforme de test

Durée estimée : 5 jours

Description :

Il s'agit de tester la migration de la version 3 du logiciel CRM Microsoft vers la version 4 et de régler les problèmes liés à cette migration aussi bien au moment de l'installation qu'après la migration. Une documentation sera élaborée afin de pouvoir effectuer la migration sur le serveur d'exploitation.

L'équipe du pôle développement devra pouvoir continuer à tester des évolutions sur la version d'exploitation pendant que les tests de la nouvelle version seront effectués. De ce fait la plateforme de test devra comporter à l'issue de cette phase à la fois la version 3 et la version 4 du logiciel de CRM.

Etape D : Analyse des processus commerciaux

Durée estimée : 60 jours

Réalisé par : GTP

Description :

Il s'agit de créer et d'animer un groupe de travail dont le but sera d'analyser les règles de fonctionnement actuelles afin d'adapter le nouveau logiciel de CRM Commercial à notre mode de fonctionnement. Ce groupe de travail devra fournir à l'issue de ses réflexions un document d'analyse présentant les processus commerciaux de la société.

Etape E : Mise en place des automatismes sur la plateforme de test

Durée estimée : 10 jours

Description :

En se basant sur le document d'analyse du groupe de travail, une retranscription des résultats de l'analyse dans l'environnement du logiciel de CRM pour la mise en place des automatismes liés aux processus commerciaux sera effectuée. Un document sur la mise en place de ces derniers devra être réalisé pour en effectuer la maintenance.

Etape F : Reprise de données sur la plateforme de test

Durée estimée : 10 jours

Description :

Il s'agit de récupérer les informations contenues dans la base de données de l'ancien logiciel commercial dans le nouveau logiciel de CRM. La rédaction d'un document sur la récupération des données sera à rédiger dans un but de maintenance.

Etape G : Création et personnalisation des écrans sur la plateforme de test

Durée estimée : 15 jours

Description :

A l'aide du document d'analyse du groupe de travail, les écrans du nouveau logiciel de CRM seront adaptés afin que toutes les informations soient disponibles. Les écrans ainsi créés seront soumis à validation auprès du groupe de travail.

Etape H : Phase de test

Durée estimée : 15 jours

Réalisé par : Chef de projet, service commercial, membre du GTP

Description :

Lors de cette phase il faut élaborer un ou plusieurs scénarios de test permettant de tester les fonctionnalités de l'application. Ces scénarios seront ensuite joués sur la plateforme de test. Une fois ces tests effectués et validés, un test de saisie en parallèle devra être fait sur la plateforme de production et celle de test pour certains utilisateurs afin de vérifier des cas d'utilisation réels.

A cette occasion une documentation utilisateur devra être rédigée afin de servir de support de formation.

Etape I : Migration SQL 2008 sur la plateforme d'exploitation

Durée estimée : ½ journée

Description :

Il s'agit de mettre en place la version SQL 2008 sur l'environnement d'exploitation en se basant sur le document élaboré lors de l'étape B.

Etape J : Migration CRM 4.0 sur la plateforme d'exploitation

Durée estimée : ½ journée

Description :

En se basant sur le document élaboré lors de l'étape décrite en C, la mise en place de la version CRM 4.0 sur la plateforme d'exploitation sera effectuée.

Etape K : Mise en place des automatismes sur la plateforme d'exploitation

Durée estimée : ½ journée

Description :

Il s'agit de mettre en place les automatismes sur la plateforme d'exploitation en se basant sur le document élaboré lors de l'étape E.

Etape L : Mise en place des écrans et des personnalisations sur la plateforme d'exploitation

Durée estimée : ½ journée

Description :

Les personnalisations créées sur la plateforme de test lors de l'étape G sont retranscrites sur la plateforme d'exploitation.

Etape M : Formation des utilisateurs

Durée estimée : 3 jours

Réalisé par : Chef de projet, représentant du service commercial au sein du GTP

Description :

Les utilisateurs sont formés à l'utilisation du nouveau logiciel de CRM en se basant sur la documentation élaborée lors de l'étape H.

Etape N : Reprise de données sur la plateforme d'exploitation

Durée estimée : ½ journée

Description :

Cette étape correspond à la reprise de données sur la plateforme d'exploitation en se basant sur le document élaboré lors de la phase F.

2.1.2 Agencement des étapes

Les étapes ayant été identifiées, elles ont été agencées les unes par rapport aux autres afin de pouvoir élaborer un diagramme de PERT (Program of Evaluation and Review Technique) servant de fil conducteur au projet.

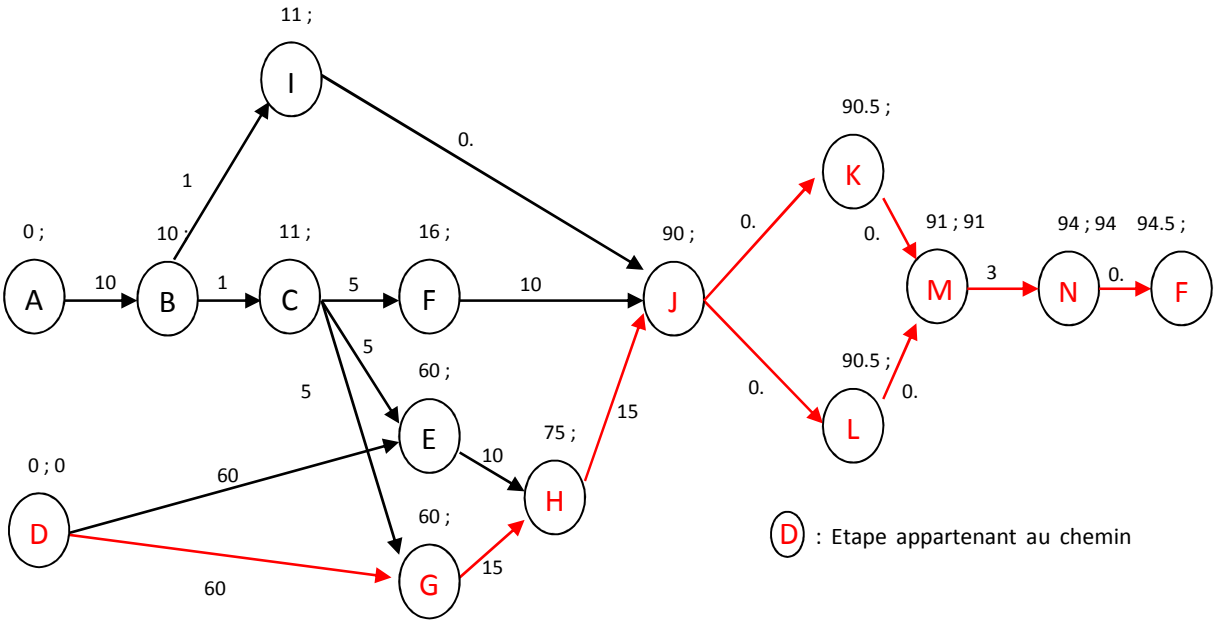


Figure 4 : Diagramme de PERT

A : Installation d'une plateforme de test
B : Migration SQL 2008 sur la plateforme de test
C : Migration CRM 4.0 sur la plateforme de test
D : Analyse des processus commerciaux
E : Mise en place des automatismes sur la plateforme de test
F : Reprise de données sur la plateforme de test
G : Création et personnalisation des écrans sur la plateforme de test
H : Phase de test
I : Migration SQL 2008 sur la plateforme d'exploitation
J : Migration CRM 4.0 sur la plateforme d'exploitation
K : Mise en place des automatismes sur la plateforme d'exploitation
L : Mise en place des écrans et des personnalisations sur la plateforme d'exploitation
M : Formation des utilisateurs
N : Reprise de données sur la plateforme d'exploitation

L'établissement du chemin critique sur ce projet m'a permis d'en dégager les étapes prioritaires, tandis que l'agencement des étapes dans le temps me permettait de déterminer sur quelles tâches je pouvais travailler entre deux réunions d'analyse. Ce diagramme et ce découpage ont ensuite été soumis à ma direction afin de servir de base pour le suivi de l'avancement du projet au travers d'un planning prévisionnel et réalisé.

2.2 Analyse

2.2.1 But de l'analyse

Lors de l'identification des tâches nécessaires à la réalisation du projet, une phase d'analyse a été planifiée. Le but de ce projet n'étant pas de créer un nouveau logiciel mais de transcrire la partie commerciale de l'ancien système dans le nouveau logiciel de CRM, cette analyse a été restreinte aux processus commerciaux. Elle devait dans un premier temps revenir sur ces processus afin de pouvoir les automatiser dans des flux de travail (Workflow) pour faciliter le travail au quotidien des différents acteurs. Dans un deuxième temps, elle devait être utilisée pour faire évoluer les processus commerciaux dans le but d'améliorer le service rendu au client tout en facilitant le travail collaboratif des différents utilisateurs.

La formalisation de ces processus devait faciliter l'intégration des nouveaux arrivants au sein du service commercial et offrir une meilleure compréhension des différentes interactions entre services. Elle devait servir également de référence pour les futurs développements et évolutions applicatives. Enfin, au travers de cette analyse, une identification des données obsolètes devait permettre d'optimiser la récupération de données.

Afin de réaliser cette étape, nous avons procédé avec ma direction à une identification des différents services acteurs de ces processus. J'avais ensuite la charge d'animer un groupe de travail projet (GTP), constitué de un ou plusieurs représentants désignés par les chefs des services concernés, dont le but était de mener à bien cette analyse. Les membres de ce groupe de travail venant d'horizons différents, et n'étant pas nécessairement rompus à l'analyse, un moyen de communication commun a été recherché.

2.2.2 Choix de la méthode et des outils

Étant donné qu'il n'y avait pas au sein de la société de méthode d'analyse définie ou faisant consensus je me suis orienté dans un premier temps sur la modélisation UML (Unified Modeling Language) [3] [4]et plus particulièrement sur les diagrammes d'activités. Au fil de mes recherches j'ai finalement orienté mon choix sur BPMN (Business Process Modeling and Notation) [5] [6] [7] qui se rapproche des diagrammes d'activité d'UML mais qui est plus adapté à la schématisation des processus. Ce choix a été consolidé par le fait que

des sociétés proposent des solutions logicielles permettant de créer des schémas BPMN générant automatiquement des flux de travail dans *Microsoft CRM*.

Afin de modéliser les processus, j'ai entrepris une prospection pour des logiciels BPMN auprès de différents éditeurs. Parmi eux nous pouvons citer AgilPoint, Aris, BonitaSoft, MetaStorm, Questetra, Progress Software, Interfacing enterprise ou encore Lombardi. Cependant au vu des contacts pris et des différents échanges commerciaux, il s'est avéré que ces solutions n'étaient pas viables financièrement à l'échelle de ce projet. Certaines de ces sociétés proposant des environnements de modélisation gratuits, ceux-ci ont été testés et finalement deux logiciels ont été retenus.

Le premier, de la société Interfacing Enterprise, permet de générer des diagrammes à partir de Microsoft Visio, ces derniers pouvant être ensuite lus grâce à la visionneuse Visio de Microsoft. Cependant il ne permet pas de gérer une documentation d'analyse.

Le deuxième, de la société Lombardi, permet de générer facilement de la documentation liée aux diagrammes BPMN mais offre moins de souplesse et de clarté dans la réalisation de ces diagrammes. De ce fait, pour modéliser les processus issus de l'analyse, les deux solutions ont été utilisées en complément l'une de l'autre.

Une fois ces choix effectués l'analyse en elle-même pouvait débuter.

2.2.3 Déroulement de l'analyse

Dans un premier temps, avec l'aide de ma direction j'ai récupéré les procédures en possession des chefs de service concernant le champ d'étude de l'analyse afin de les schématiser en utilisant les outils BPMN. Cette étape avait pour but de créer une base de travail tout en me permettant de me familiariser avec les outils et la notation BPMN. A l'issue de cette phase j'ai élaboré un document de présentation de la notation BPMN qui a été transmis aux membres du GTP [**Voir annexe 1**]. Ce document a été présenté et discuté lors de la première réunion du GTP afin que chacun des membres intègre la notation BPMN. Lors de cette réunion le but de l'analyse ainsi que la méthode de travail ont été présentés aux membres du GTP.

Par la suite, en accord avec les membres du GTP, une réunion hebdomadaire de deux heures a été planifiée sur une période de dix semaines durant laquelle une analyse des

processus commerciaux devait être effectuée. La schématisation des processus transmis par les chefs de service servait de base de départ à cette analyse. A l'issue de chaque réunion je rédigeais un compte-rendu [**Voir annexe 2**] transmis ensuite aux membres du GTP et je mettais à jour les processus en fonction des remarques effectuées. Ces modifications étaient ensuite validées lors de la réunion suivante par les membres du GTP. En parallèle de ces comptes rendus je maintenais à jour une schématisation globale des processus commerciaux ainsi qu'une documentation afin d'élaborer le document d'analyse.

2.3 Récupération de données

2.3.1 Considérations sur la récupération de données

La première difficulté dans la reprise de données réside dans la répartition de ces données au sein de trois applications distinctes, chacune contenant des données parfois redondantes.

En effet, le logiciel *Microsoft CRM* ne servant jusqu'à aujourd'hui qu'au suivi clients au niveau des incidents hotline, ce dernier ne contient que la liste des clients susceptibles d'être appelés ou de déposer un incident. Le logiciel alfa.tiers contient quant à lui l'ensemble des informations administratives du fichier clients mais sans les informations purement commerciales. Enfin le logiciel *Selligent* contient l'ensemble du fichier clients avec les informations commerciales.

Historiquement les informations pouvaient être modifiées au sein de ces trois applications de manière indépendante sans qu'il n'y ait de processus répercutant les modifications au sein des autres applications. Une consolidation a été effectuée depuis et le point d'entrée a été défini comme étant le logiciel alfa.tiers. Ce dernier contient donc les mises à jour les plus récentes et les répercute au sein des deux autres applications. Cependant des incohérences de données peuvent tout de même subsister. Pour effectuer la récupération de données des règles ont donc été définies en concertation avec le service commercial:

- Les données administratives contenues dans le logiciel alfa.tiers sont considérées comme étant à jour et faisant foi pour l'ensemble des autres applications.

- Les fiches clients éventuellement présentes dans le logiciel *Selligent* mais absentes du logiciel alfa.tiers feront l'objet d'un traitement manuel au cas par cas.
- Les données administratives du logiciel *Microsoft CRM* doivent être le reflet exact du logiciel alfa.tiers
- Les données et informations commerciales seront récupérées du logiciel *Selligent* et incorporées dans le logiciel de CRM pour les fiches clients présentes dans le logiciel alfa.tiers.

La deuxième difficulté de cette reprise de données consiste dans le volume des données à traiter. En effet les logiciels *Microsoft CRM* et *Selligent* contiennent chacun un volume de données de 18 Go représentant dans certaines tables de *Selligent* plus de 300 000 entrées et plus de 4 millions d'entrées dans certaines tables de *Microsoft CRM*.

Ces règles de gestion ainsi que ces volumes importants de données ont donc dû être pris en considération lorsque nous avons fait le choix, avec ma direction, d'une méthode d'importation de données.

2.3.2 Méthodes d'importation

Trois méthodes se sont imposées naturellement à nous pour effectuer l'importation de données.

La première d'entre elles est l'utilisation du programme d'importation Data Migration fourni par Microsoft pour importer les données dans le logiciel de CRM à partir d'un fichier au format csv. La seconde est d'injecter les données directement dans la base de l'application du logiciel de CRM via des requêtes SQL. Enfin la dernière est la création d'une application utilisant le SDK de Microsoft pour importer les données.

Nous allons voir quels sont les avantages et les inconvénients de chacune de ces méthodes afin de comprendre ce qui a orienté notre choix.

2.3.2.1 Logiciel d'importation Microsoft Data Migration

Il s'agit de la méthode la plus simple et demandant le moins de développement. En effet le logiciel fourni permet au travers d'un assistant d'importer facilement des données à partir d'un fichier au format csv.

Le fichier doit être formaté et les colonnes doivent porter le nom des champs du logiciel *Microsoft CRM* pour que la liaison entre les données puisse se faire automatiquement. La transformation des types de données entre le fichier source et la destination se fait quant à elle, de manière automatique pour les types simples. Il suffit alors de créer une requête récupérant les données de l'ancien système et d'exporter le résultat dans un fichier csv.

Cependant les tests effectués ont montré certaines limitations contraignantes du fait de l'environnement et des méthodes de travail de la société.

En effet, lors de l'importation, le compte utilisateur servant à effectuer l'importation devient titulaire des enregistrements importés. Or dans les méthodes de travail de la société, la notion de titulaire définit le portefeuille des commerciaux et des assistantes commerciales.

Une deuxième contrainte réside dans la limitation à 8 000 lignes d'importation à la fois. Au-delà de ce nombre l'application doit être exécutée plusieurs fois, ce qui, au vu du nombre de lignes dans certaines tables de *Selligent*, n'est pas viable.

2.3.2.2 SQL

Dans cette méthode il s'agit de créer des requêtes SQL. Ces dernières vont récupérer les informations que l'on souhaite importer dans la base de l'application *Selligent* et les injecte directement dans la base de données de l'application *Microsoft CRM*. Les types de données de la source doivent être transformés au sein de la requête afin de les rendre compatibles avec la destination. Les différentes contraintes des bases de données de destination doivent être respectées.

Cette méthode est sans doute la plus rapide en temps de traitement. Cependant elle nécessite une parfaite connaissance de la structure des bases de données des applications *Selligent* et *Microsoft CRM*. Sans cela des erreurs peuvent être commises et engendrer une instabilité de l'application de destination.

Cette méthode n'est pas approuvée par Microsoft qui ne fournit donc pas de modèle de la structure de sa base de données, considérant que l'accès aux données doit se faire par le modèle objet fourni par le SDK.

2.3.2.3 Création d'une interface d'importation

Cette dernière méthode consiste à supprimer les limitations rencontrées avec le logiciel Microsoft Data Migration. Il s'agit de créer une application permettant de récupérer les données de l'ancien système et de les intégrer dans *Microsoft CRM* en s'appuyant sur le SDK fourni par Microsoft.

Le fait de passer par le SDK permet d'assurer l'intégrité des données insérées dans le nouveau système et de procéder à des transformations des types de données de *Selligent* pour les rendre compatibles avec ceux de *Microsoft CRM*.

Cette solution laisse également plus de flexibilité dans la reprise de données car il est à ce moment-là possible de définir le propriétaire des différentes fiches et de ne plus être limité par le nombre d'enregistrements simultanés pouvant être créés.

Elle est cependant plus longue à mettre en place du fait qu'elle nécessite la création d'une application complète permettant de faire la reprise de données.

2.3.3 Choix de la méthode de récupération de données

Au vu des différentes méthodes évoquées nous avons décidé de créer une interface de récupération de données.

Cette dernière s'inspirera de l'application Microsoft Data Migration dans l'enchaînement des étapes mais proposera des contrôles d'anomalies spécifiques à notre implémentation. La récupération de données se fera au travers de requêtes SQL interrogeant directement la base de données de l'application *Selligent*. Ces données seront ensuite injectées dans l'application *Microsoft CRM* en utilisant le modèle objet de l'application et notamment ses services Web.

De cette manière nous pourrons attribuer les différentes fiches aux propriétaires définis dans *Selligent* et conserver un certain nombre d'informations comme par exemple les dates de création de document.

J'ai fait le choix de rendre cette application la plus générique possible et faiblement liée aux données sources afin de pouvoir modifier facilement les requêtes SQL servant de source

d’importation. Le but était de permettre ultérieurement la réutilisation de cette application pour importer des données de sources diverses.

A l’issue de la définition de cette méthodologie, nous allons pouvoir nous attacher à décrire la réalisation de ce projet au travers de son application.

Chapitre 3 Travail réalisé

3.1 Installation d'une plateforme de test

Afin de pouvoir effectuer les tests tout au long du projet, j'ai mis en place une plateforme constituée d'un serveur Microsoft Windows 2008 avec les services Hyper-v activés. Cette plateforme, en plus de servir pour le projet, venait en remplacement de l'ancienne plateforme de développement utilisée par l'équipe de développement de la gestion interne.

J'ai fait le choix d'utiliser des machines virtuelles pour la plateforme de test et de développement afin de bénéficier de la possibilité d'effectuer des captures instantanées (snapshot) des postes et de les restaurer rapidement. De plus cette technologie permettait de limiter le nombre de postes physiques et de créer facilement des réseaux séparés des réseaux d'exploitation.

Cette plateforme est constituée de plusieurs serveurs virtuels reflétant la configuration de production (voir Figure 5 p.29). Elle comprend :

- deux contrôleurs de domaine duplicata des contrôleurs de production.
- deux serveurs de messagerie duplicata de ceux de production.
- un serveur Web duplicata de celui de production.
- un serveur SQL installé à l'identique de celui de production et comprenant une copie des bases de production.
- une station de travail sous Windows XP servant de client et ayant un environnement de développement installé en Visual Studio 2008.

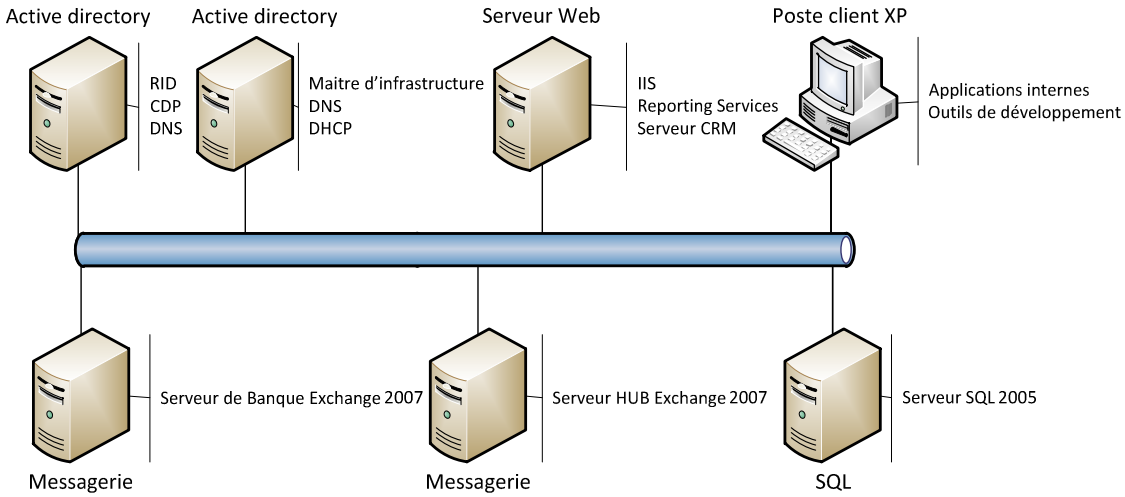


Figure 5 : Architecture de la plateforme de test

Une fois la configuration de base terminée un snapshot de tous les postes de la plateforme de test a été effectué.

3.2 Migration SQL 2008 sur la plateforme de test

Lors de cette phase, j'ai récupéré les anomalies remontées par le logiciel d'installation de SQL 2008 et je les ai transmises à l'équipe de développement de la gestion interne afin qu'elles soient corrigées sur l'environnement de test et de production. J'ai ensuite testé les différentes applications liées aux bases de données présentes sur le moteur SQL à partir du poste client de la plateforme de test afin de détecter d'éventuelles anomalies de fonctionnement. Le service de rapport : Reporting Services, présent sur le serveur Web utilisé pour les rapports internes et par *Microsoft CRM* a été mis à jour à cette occasion.

J'ai rédigé, lors de cette phase de test, une procédure pas à pas détaillée permettant d'effectuer ce changement de version. Une fois cette étape terminée un snapshot a été remonté et la procédure a été suivie pas à pas afin de valider son contenu.

Pour la suite du traitement un snapshot a été remonté et une nouvelle instance SQL 2008 a été mise en place en parallèle de l'instance SQL 2005. Le but de cette configuration était de permettre de continuer à faire les tests sur la plateforme SQL 2008 tout en offrant à l'équipe de développement la possibilité de travailler dans un environnement similaire à celui de production.

3.3 Migration CRM 4.0 sur la plateforme de test

3.3.1 Migration du logiciel

Il s'agissait lors de cette phase de tester la migration de la version 3 du logiciel de CRM vers la version 4 à partir des kits d'installation fournis par Microsoft.

A l'instar de la migration SQL, j'ai rédigé une procédure pas à pas détaillée lors de cette phase de test de migration afin de pouvoir l'effectuer ultérieurement sur la plateforme de production en évitant les écueils rencontrés lors de cette étape.

Une fois la procédure rédigée un snapshot a été remonté et j'ai effectué de nouveau la migration en suivant scrupuleusement la procédure afin d'en valider le contenu. Ultérieurement, à chaque mise à jour majeure de la procédure globale pouvant avoir un impact sur le processus de migration, je rejouais le scénario de migration sur la plateforme de test.

Pour la suite du traitement j'ai mis en place un serveur Web supplémentaire afin d'avoir sur la plateforme de test une version *Microsoft CRM* 4.0 et *Microsoft CRM* 3.0 respectivement pour les tests et le maintien de la plateforme de production.

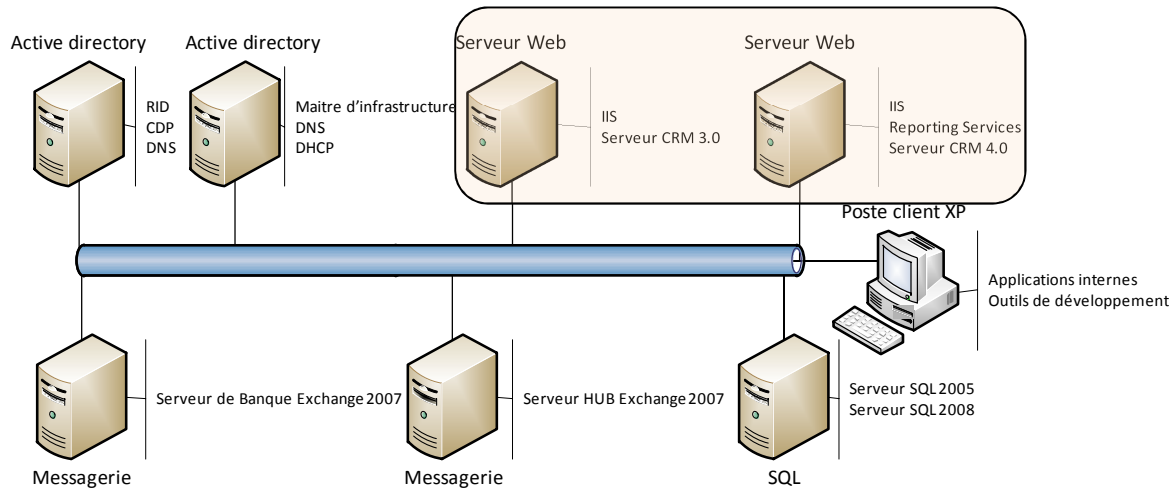


Figure 6 : Nouvelle architecture de la plateforme de test

3.3.2 Modification des personnalisations

Une fois la migration logicielle effectuée, j'ai testé les fonctionnalités de la nouvelle plateforme et notamment les personnalisations afin de valider leur fonctionnement sur la nouvelle version. Ces tests ont révélé des dysfonctionnements qui m'ont amené à effectuer des modifications. Celles-ci avaient un impact à la fois sur les scripts et sur les développements .Net.

3.3.2.1 Les scripts

A l'origine les scripts étaient inclus directement dans les pages du logiciel de CRM au travers de l'interface de personnalisation des formulaires de l'application. De ce fait, pour connaître le script s'exécutant pour une entité donnée, il fallait se positionner sur le formulaire de l'entité, l'éditer, puis accéder à ses propriétés. Ces dernières permettaient de vérifier le contenu des scripts de chargement et de sauvegarde comme le montre la Figure 7. De plus, chaque zone de saisie du formulaire pouvait avoir dans ses propriétés ses propres scripts.

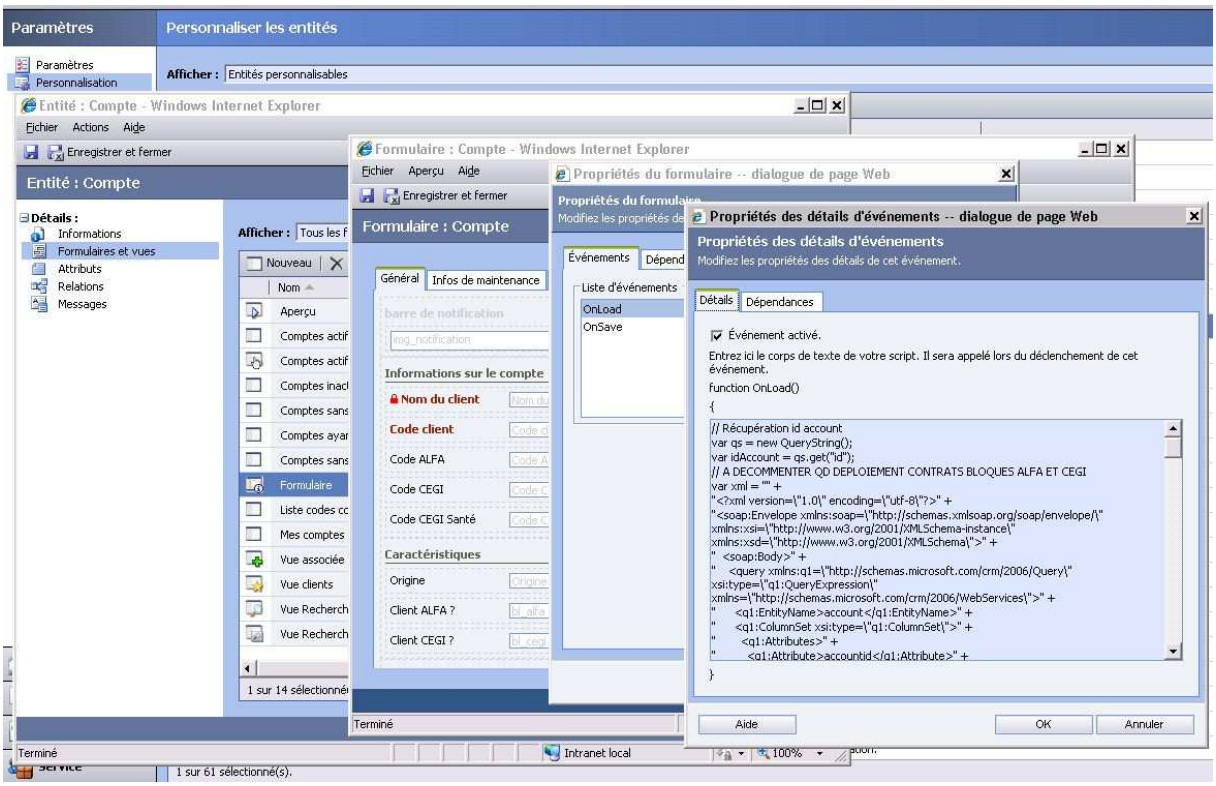


Figure 7 : Ensemble des écrans permettant d'accéder au script de chargement du formulaire de l'entité Compte dans *Microsoft CRM* 3.0

De par cette configuration certains de ces scripts étaient répliqués à l'identique au sein des différents formulaires rendant la maintenance complexe. Ils étaient également rechargés à chaque ouverture de formulaire puisqu'ils étaient directement inclus dans leurs propriétés.

Afin d'améliorer les temps de réponse et de simplifier la maintenance et l'évolution de ces scripts, j'ai décidé de les externaliser au sein de deux fichiers, l'un pour la gestion évènementielle et l'autre servant de bibliothèque de fonction pour le premier. J'ai ensuite supprimé les scripts des formulaires et des zones de saisie. Enfin, j'ai remplacé le script de démarrage de chaque formulaire par un simple appel au fichier évènementiel qui prenait en paramètre le nom du formulaire.

Cette modification a permis de structurer et d'épurer le code des scripts des redondances présentes. De plus les fichiers sont maintenant chargés dans le cache du navigateur lors du premier appel d'un formulaire. Par la suite le cache est utilisé par les autres formulaires afin d'améliorer les temps de chargement.

Une fois ces modifications structurelles effectuées j'ai retranscrit le code pour le rendre compatible avec la version 3 du logiciel de CRM afin d'inclure ces modifications en amont de la migration et simplifier la procédure de migration. La procédure de migration a été alors modifiée pour inclure ce travail préparatoire et le scénario de migration a été rejoué sur la plateforme de test.

3.3.2.2 Les développements .Net

En cherchant à corriger les anomalies constatées lors de la migration en version 4 du logiciel de CRM j'ai été amené à étudier les développements .Net utilisés pour les personnalisations. Ces derniers étaient regroupés au sein d'une solution de personnalisations regroupant quatre projets interconnectés. Ces projets faisaient appel à des services Web ou étaient eux-mêmes appelés par des services Web. Le schéma suivant représente ces projets et leurs liens d'interconnexion.

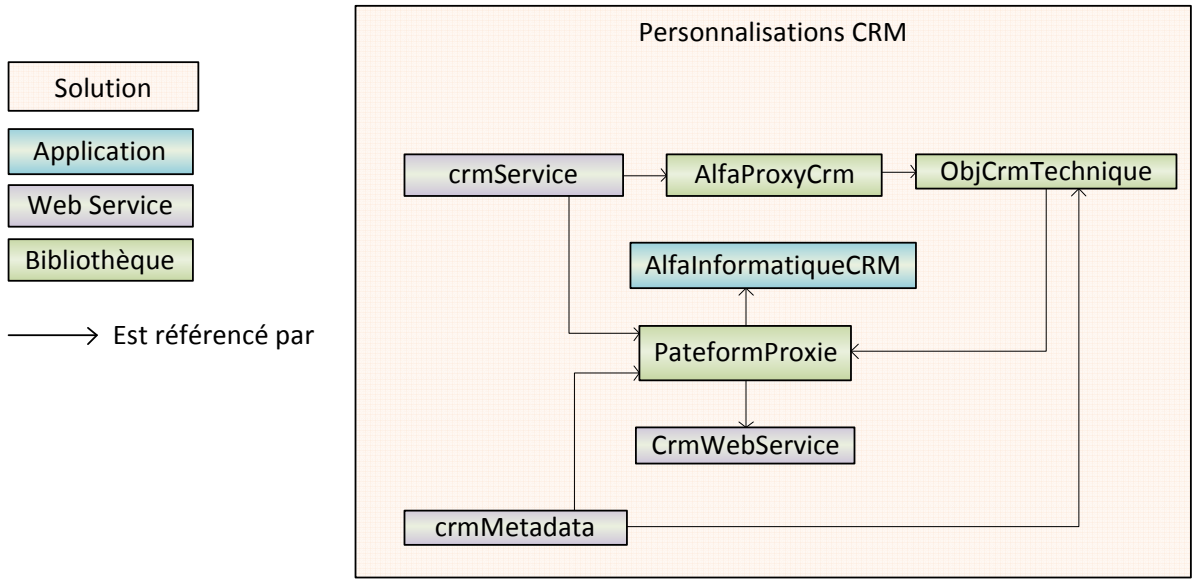


Figure 8 : Liaisons entre projets au sein de la solution des personnalisations du logiciel de CRM

Le projet « AlfaInformatiqueCRM » peut être assimilé à la couche présentation de ces développements. Elle regroupe l'ensemble des pages personnalisées mises à disposition des utilisateurs. Ce projet était composé de 48 pages aspx, chacune étant indépendante des autres et ayant son propre code. J'ai pu constater que ce code avait été fait à différentes époques, à l'aide des technologies disponibles alors, par différents développeurs mais que globalement certaines pages du projet étaient identiques.

J'ai donc entrepris un travail de consolidation de ces pages en utilisant les outils mis à disposition par le Framework 3.5 de Microsoft à savoir les extensions et les délégués. De ce fait j'ai pu réduire le nombre de pages aspx à 19 en les rendant plus génériques. Cette diminution du nombre de pages a permis de réduire la complexité de la maintenance et ainsi

limiter le nombre de modifications liées au changement de version du logiciel de CRM. De plus, le nombre d'objets sollicités étant moindre, les objets encore présents sont chargés plus longtemps et plus fréquemment en mémoire. Cela permet ainsi de diminuer les temps de chargement tout en diminuant la charge mémoire. En complément, les pages faisant des interrogations de bases de données remontant un nombre important d'enregistrements ont été modifiées afin d'utiliser la mise en cache des enregistrements pour limiter le nombre de requêtes effectuées sur le moteur SQL.

Le projet « AlfaProxyCrm » était utilisé pour améliorer les performances du logiciel de CRM et des applications tierces nécessitant un accès aux services Web du logiciel de CRM. Il s'agissait de créer une dll (Dynamic Link Library) du service Web fourni par Microsoft ; après avoir épuré du code de ce dernier celui qui n'était pas utilisé dans les applications tierces. De cette manière, la dll était chargée plus rapidement. Cependant cette solution manquait de flexibilité. En effet le service Web auquel faisait référence la dll contenait la définition des entités du logiciel de CRM et de leurs attributs. De ce fait, à chaque rajout d'un attribut dans le logiciel de CRM, la dll devait être de nouveau générée, épurée puis recompilée avant d'être réintégrée dans les différents projets la référençant afin de pouvoir utiliser ces nouveaux champs du logiciel de CRM Microsoft. Cette méthode était en place depuis les premières versions du logiciel de CRM. J'ai donc effectué de nouveaux tests avec la version 3 et la version 4 de *Microsoft CRM* en référençant directement dans les projets le service Web concerné. Les tests de chargement effectués ont montré un écart de l'ordre de la seconde dans cette configuration par rapport à l'ancienne méthode. Du fait du gain de flexibilité et du faible écart de temps de chargement, nous avons décidé avec l'équipe de développement de la gestion interne de référencer directement les services Web du logiciel de CRM et de ne plus utiliser le projet « AlfaProxyCrm ».

Les projets « ObjCrmTechnique » et « PlateformProxie », quant à eux, avaient pour rôle de fournir un ensemble de méthodes permettant d'accéder aux objets du logiciel de CRM Microsoft mis à disposition par les services Web de l'application. Ils représentaient donc la couche métier des applications internes et permettaient l'interfaçage de ces applications avec le logiciel de CRM Microsoft. Le projet « PlateformProxie » n'était utilisé que dans la solution des personnalisations du logiciel *Microsoft CRM* et fournissait quelques méthodes spécifiques, notamment des méthodes d'authentification. Le projet « ObjCrmTechnique »,

quant à lui, était référencé dans toutes les applications internes ayant un lien avec le logiciel de CRM Microsoft. Afin d'anticiper la migration de la partie commerciale vers la solution *Microsoft CRM*, les deux projets ont été fusionnés pour permettre une interaction plus forte entre les projets internes et le logiciel de CRM de Microsoft. Cette nouvelle organisation a permis de simplifier l'architecture du projet des personnalisations comme le montre la représentation suivante.

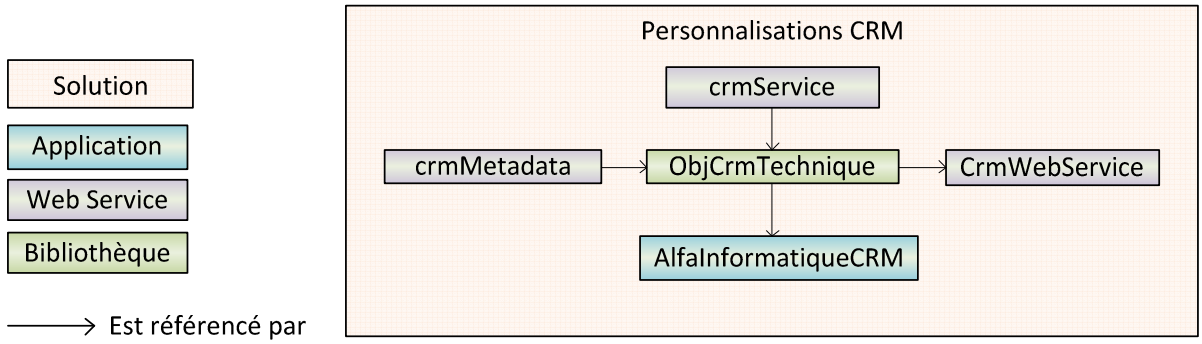


Figure 9 : Nouvelle organisation des projets au sein de la solution des personnalisations du logiciel de CRM

A la suite de ces modifications structurelles, j'ai rectifié la procédure de mise à jour afin de les intégrer en amont de la migration. J'ai ensuite de nouveau testé cette dernière.

Bien que ces modifications n'aient pas été planifiées au début du projet et que je leurs ai consacré du temps, elles se sont révélées nécessaires. En effet, elles m'ont permis de me familiariser avec les outils de développement et les objets spécifiques du logiciel de CRM Microsoft. J'ai pu ainsi m'approprier le code en le structurant, ce qui a eu pour effet de me faire gagner du temps par la suite ; notamment lors de la personnalisation des écrans et de la création de l'application de reprise de données.

3.4 Analyse des processus commerciaux

Lors du premier travail d'analyse effectué avec les membres du GTP trois processus principaux sont apparus, ces derniers étant logiquement conformes à tout processus commercial, à savoir le processus d'avant-vente, de vente et d'après-vente. Par la suite, au fil des différentes réunions, un travail itératif a permis de détailler le processus d'avant-vente et une partie du processus de vente. Le processus d'après-vente, quant à lui, a été esquissé grâce aux procédures transmises par les membres du groupe de travail.

Au fil des réunions, une demande de plus en plus forte des participants a été exprimée concernant l'absence de maquette permettant de voir la concrétisation de l'analyse. De plus, certains membres ont fait remarquer qu'ils préféreraient une retranscription telle quelle de l'ancien système dans le nouveau logiciel de CRM pour se servir de cette base de travail pour la suite.

Afin de désamorcer les risques de polémique, il a donc été décidé par la direction du service de repousser l'analyse à une date ultérieure et de profiter de la période des congés estivaux pour retranscrire l'analyse déjà effectuée au sein du nouveau logiciel de CRM. Par la même occasion, l'ancien logiciel de CRM a été transposé dans le nouveau afin d'avoir un support visuel pour la suite.

3.5 Création et personnalisation des écrans sur la plateforme de test

Suite à la décision de reporter l'analyse, j'ai commencé à intégrer une partie du résultat de l'analyse au sein du logiciel de CRM de Microsoft sur la plateforme de test. J'ai ensuite repris les différents écrans du logiciel *Selligent* et j'ai retranscrit les champs les plus pertinents dans leurs pendants au sein du logiciel de CRM de Microsoft.

Une fois ce travail effectué, j'ai organisé des réunions en groupe restreint avec les représentants du service commercial faisant partie du GTP afin de valider ces personnalisations avant leur présentation en session plénière du GTP.

En parallèle de ce travail, toutes les modifications étaient notées afin de pouvoir être retranscrites sur la plateforme de production. De ce fait j'ai rédigé une procédure post-migration incluant la mise en place des personnalisations. Une fois les modifications terminées, j'ai rejoué la procédure afin d'en valider chaque étape.

3.6 Reprise de données sur la plateforme de test

Afin d'effectuer la reprise de données, j'ai créé un logiciel s'appuyant sur le modèle objet du logiciel de CRM Microsoft et reprenant une partie des concepts du logiciel d'importation Microsoft Data Migration.

Il s'agit d'une application avec une interface Web. L'interface a pour but de suivre étape par étape l'intégration des données au travers d'une série de contrôles faits en amont du traitement.

Le traitement en lui-même se fait au travers d'un projet .Net regroupant des classes permettant d'utiliser le modèle objet du logiciel de CRM et d'effectuer des interrogations de bases de données. J'ai choisi de m'appuyer sur le modèle objet fourni par les services Web 2007 du logiciel de CRM Microsoft afin de rendre l'application compatible avec des versions futures de ce logiciel. Cependant, cette décision empêche l'application de fonctionner telle quelle avec la version 3.0 du logiciel de CRM, ce qui implique d'avoir effectué la migration en *Microsoft CRM* 4.0 avant la reprise de données.

La récupération des anciennes données se fait au travers de requêtes SQL stockées dans une base de données propre à l'application et remontant les informations présentes dans les bases des applications *Selligent* et Alfa.tiers.

La liaison avec les champs ainsi concernés et ceux du logiciel de CRM se fait grâce à des alias sur les colonnes des requêtes SQL ; ceux-ci correspondant au nom des champs dans le logiciel de CRM de Microsoft.

Une trace des clés, du logiciel de CRM Microsoft et de *Selligent*, qui correspondent aux données intégrées ou mises à jour, est conservée au sein d'une table de correspondance dans la base de données de l'application. Celle-ci contient les identifiants sources, destinations, l'entité concernée et la date de création dans cette table. Elle permet ainsi de retrouver les informations créées ou mises à jour dans le logiciel de CRM et de les comparer avec celles d'origine.

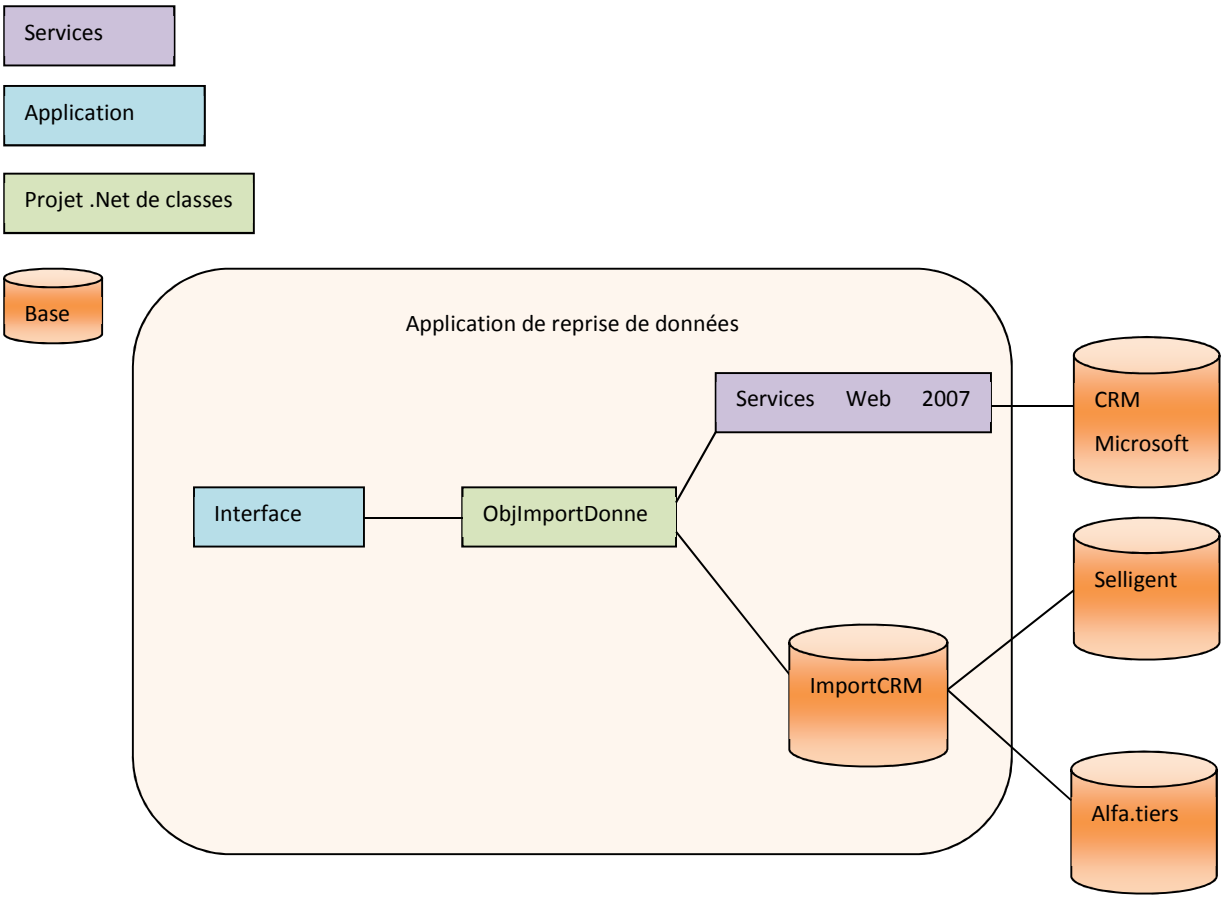


Figure 10 : Schéma simplifié du logiciel de reprise de données

J'ai fait le choix d'une interface Web dans l'optique d'intégrer l'application à postériori dans le logiciel de CRM Microsoft. J'ai choisi de procéder, comme pour le logiciel Microsoft Data Migration, par étape et par type d'entité.

L'importation concerne les entités suivantes : entité juridique, société, prospects, personnes, opportunité, actions, campagnes.

L'analyse n'étant pas encore terminée au moment de la création de ce logiciel, les données devant être récupérées n'étaient que partiellement identifiées. De ce fait, j'ai choisi de faire une application la plus générique possible et peu dépendante de la source des données afin de pouvoir rajouter facilement les champs supplémentaires à récupérer une fois l'analyse terminée.

En poussant un peu plus loin cette généralisation, j'ai fait le choix de faciliter l'importation de sources différentes, l'intérêt étant de pouvoir réutiliser ce programme pour importer des données autres que celles contenues dans *Selligent*. Dans ce but, la seule liaison étant faite avec les applications tierces se situe au niveau des requêtes SQL contenues sous forme de vues dans la base de l'application. Ainsi, en cas de modification de la source, il suffit de modifier ces vues au niveau SQL.

Cinq étapes découlent de cette généralisation de l'importation des données. Chacune de ces étapes est associée à un écran de l'application :

- Importation de données (voir Figure 11) : Il s'agit de remonter de manière brute les informations devant être importées dans le logiciel de CRM Microsoft. Cette étape permet de contrôler le nombre d'enregistrements devant être traités et de définir la clé primaire de la requête source. Cette clé primaire sera ensuite utilisée pour connaître les enregistrements ayant été traités. Elle sera utilisée comme identifiant source dans la table de correspondance.

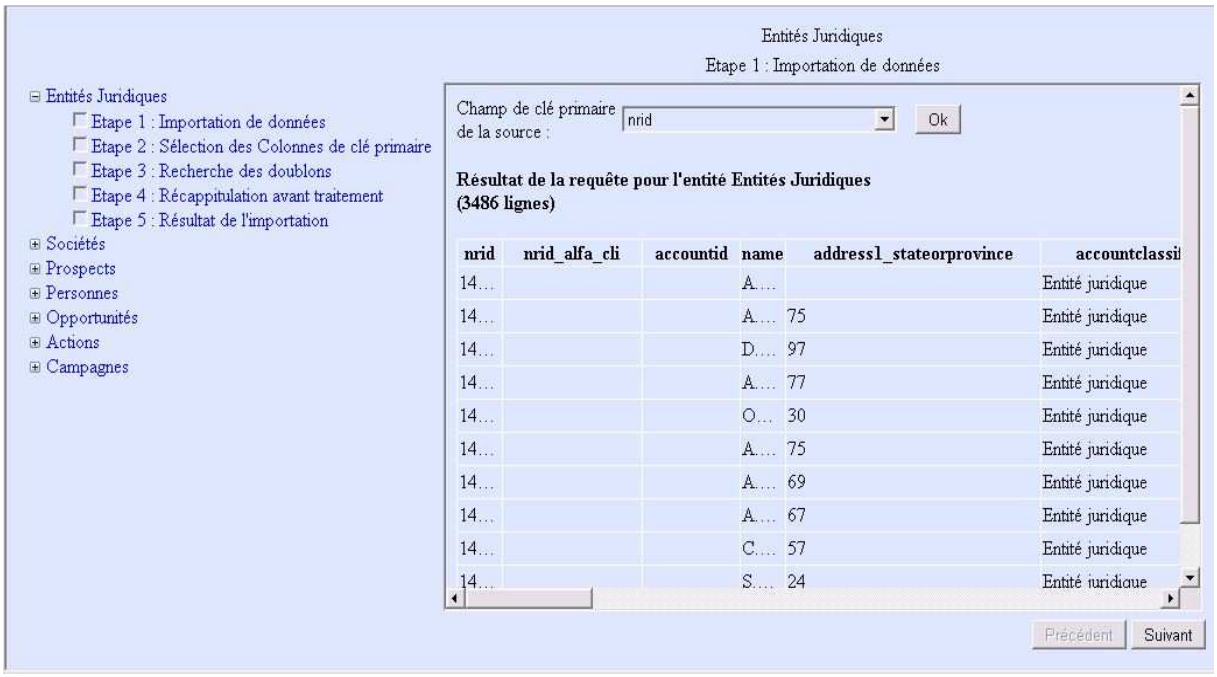


Figure 11 : Etape 1 - Importation de données

- Sélection des colonnes de clé primaire (voir Figure 12) : Lors de cette étape une corrélation entre les colonnes de la requête source et les champs du logiciel de CRM a lieu. Celle-ci se fait en reliant les noms de colonne définis dans la requête SQL avec ceux des champs présents dans le logiciel de CRM Microsoft pour l'entité traitée. Le résultat de cette corrélation apparaît dans la zone « Colonnes disponibles ». La sélection d'un champ dans la zone « Colonnes de clé primaire » permet de déterminer, parmi ces champs, lesquels permettront d'identifier les enregistrements déjà présents dans le logiciel de CRM Microsoft et donc ceux nécessitant d'être mis à jour et non pas créés. Lors de cette étape, la mise à jour des champs utilisateurs peut être forcée. En effet, par défaut dans le cadre d'une mise à jour d'enregistrement, ces champs ne sont pas modifiés s'ils sont renseignés afin d'éviter de faire une mise à jour à partir de données obsolètes.

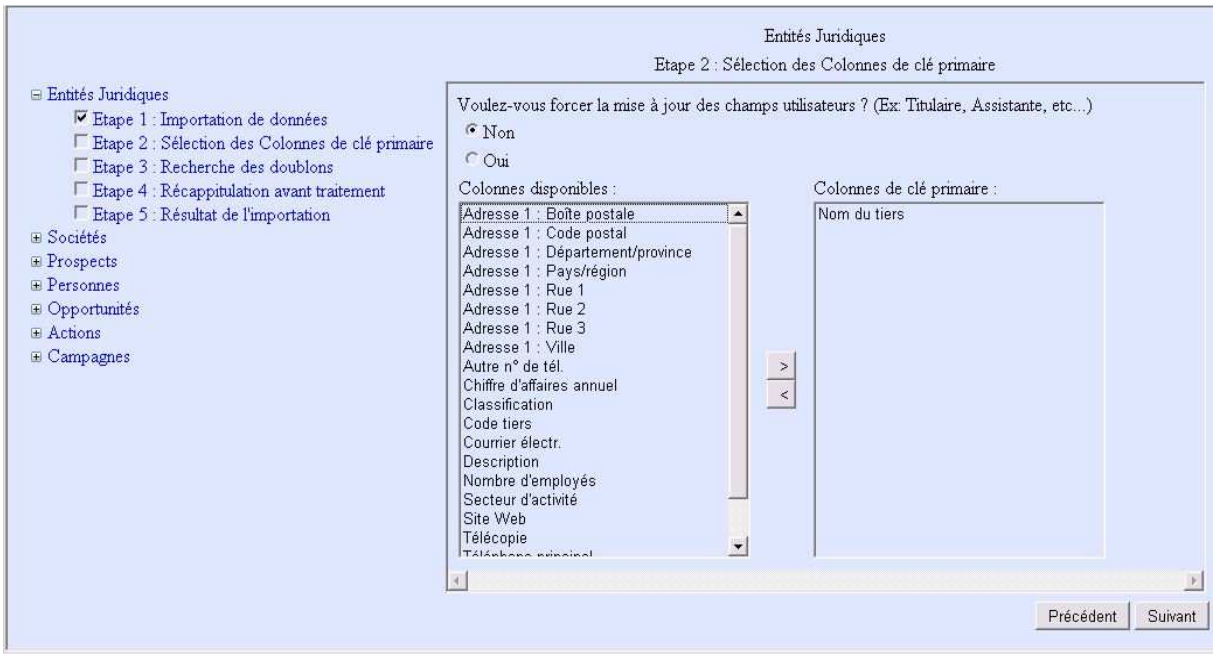


Figure 12 : Etape 2 - Sélection des colonnes de clé primaire

- Recherche des doublons (voir Figure 13) : En se basant sur les « Colonnes de clé primaire » renseignées dans l'étape précédente, une recherche de doublons est effectuée sur les données de la source et de la destination. Ces enregistrements doivent alors être traités au cas par cas en amont du traitement d'importation. La liste peut être exportée sous forme de fichier csv afin d'être transmise aux services concernés. Le service commercial doit

alors déterminer s'il s'agit bien de doublons et le cas échéant faire une fusion de ces enregistrements.

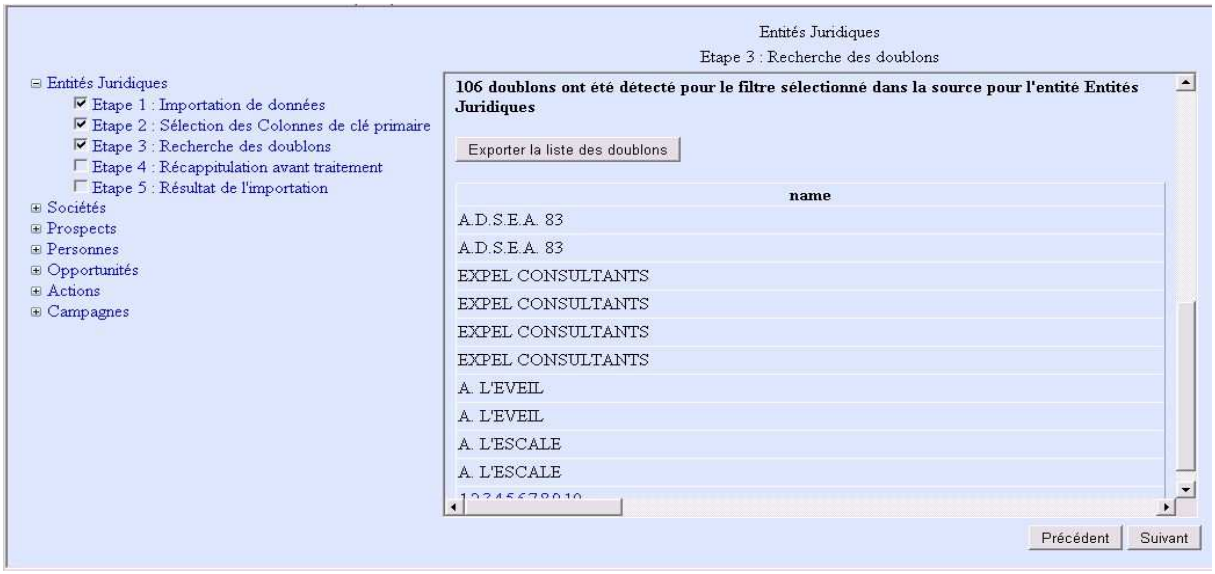


Figure 13 : Etape 3 - Recherche des doublons

- Récapitulation avant traitement (voir Figure 14) : En s'appuyant sur les clés primaires définies lors de la seconde étape, les enregistrements sont séparés en deux catégories. Ceux présents dans le logiciel de CRM Microsoft et devant donc par là-même être mis à jour et ceux n'étant pas présents et devant donc être créés. Ces deux listes peuvent être exportées au format csv pour effectuer un contrôle si nécessaire et garder une trace du traitement devant être effectué.

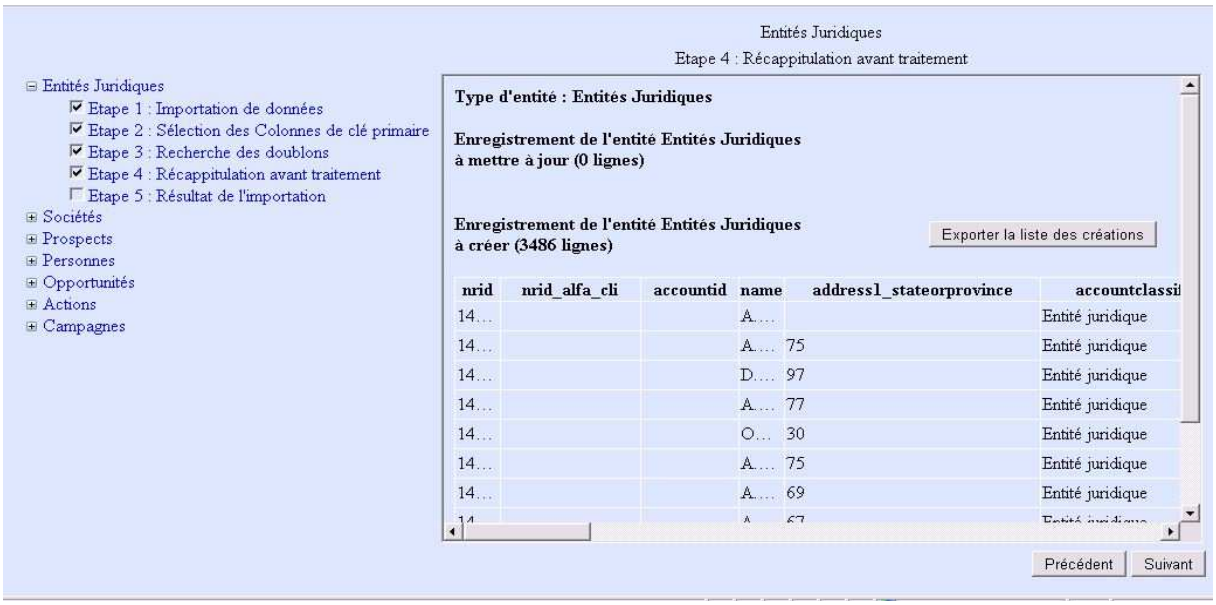


Figure 14 : Etape 4 - Récapitulatif avant traitement

- Résultat de l'importation (voir Figure 15) : Une fois le traitement effectué un récapitulatif est affiché. En cas d'erreur lors du traitement un journal des erreurs sous forme de fichier est créé et un bouton permettant d'accéder à ce dernier apparaît.



Figure 15 : Etape 5 - Résultat de l'importation

Ce logiciel étant un utilitaire destiné dans un premier temps aux seuls administrateurs du logiciel de CRM Microsoft et de *Selligent*, j'ai préféré me concentrer sur la partie fonctionnelle et la stabilité de l'application au détriment de l'interface graphique. Pour les mêmes raisons certaines modifications, telles que la modification des requêtes permettant la récupération de données, nécessitent une connaissance des requêtes SQL et de la structure des bases de données des applications.

3.7 Ecart entre le planning prévisionnel et réalisé

De nombreux éléments ont eu un impact direct sur le planning initial. En effet, avant de commencer certaines étapes, j'ai eu à me former sur l'environnement et le langage de programmation. J'ai dû également rechercher des logiciels et des méthodes d'analyse et me familiariser avec le nouveau logiciel de CRM. Ces étapes n'apparaissaient pas au niveau du planning initial car elles ne faisaient pas partie directement du projet. Elles ont été prises en compte par la suite car le temps leur étant accordé, en l'occurrence 25 jours, n'était pas négligeable à l'échelle du projet.

D’autres impondérables sont venus rallonger les temps initialement définis. Par exemple l’installation de la plateforme de test a nécessité 5 jours de travail supplémentaires en raison de problèmes matériel et de volume de stockage.

L’étape de migration CRM 4.0 sur la plateforme de test a été plus longue de 5 jours. Cette étape comportait une phase de premier test de compatibilité des applications tierces. Or l’identification des problèmes rencontrés et la définition des mesures correctives à apporter a été plus longue que prévue.

L’analyse des processus a été écourtée à 40 jours. La raison principale est la non prise en compte de la période de congé estivale. Cette dernière a rendu difficile la réunion des membres du groupe de travail. Les jours restant ont été reportés à une date ultérieure d’un commun accord. Cette période a tout de même permis de formaliser les processus d’avant-vente et de vente ainsi que de rédiger l’analyse de ces processus. Ces derniers ont pu être schématisés grâce aux documents fournis et à la tenue de 6 réunions plénières du groupe de travail.

La période de congé a été mise à profit pour revenir sur la version 4 de *Microsoft CRM.* Ce retour était nécessaire de par le nombre d’erreurs rencontrées sur les personnalisations, ces dernières étant liées au changement de version. Le nombre de modifications à apporter, ainsi que l’éparpillement du code m’ont conduit à essayer d’optimiser le code existant en le regroupant et en l’adaptant à la version 4 de *Microsoft CRM.* Ce travail était nécessaire en vue des futures évolutions du logiciel de *Microsoft CRM*. Cependant cette étape nécessitait la lecture entière du code des personnalisations, sa correction et des tests de non régression. Elle a donc été couteuse en temps et a retardé le planning initial de 60 jours.

Lors de l’établissement du premier planning, il était pressenti que nous pourrions utiliser le logiciel *Microsoft Data Migration* pour la reprise de données de l’ancien logiciel de CRM. Le planning avait donc été établi en s’appuyant sur une estimation de la phase de test de ce logiciel. Cependant les tests ont démontré la nécessité de créer un logiciel de reprise de données. Cette opération a nécessité une durée de 40 jours.

L’ensemble de ces contretemps m’a amené à concevoir différemment l’approche de la création d’un planning de projet en tenant compte du facteur risque. Ces risques peuvent

être de différentes nature : humaine telle les congés, matérielle ou liée à des facteurs d'incertitudes tels la nécessité de créer un logiciel de reprise de données. Ces éléments m'ont permis d'avoir un retour d'expérience sur la gestion du facteur temps dans la conduite de projet.

Chapitre 4 Retour d'expérience

« L'expérience, c'est ce qui nous permet de reconnaître une erreur quand nous la recommençons. » [Franklin P. Jones]. Cette citation pourrait être complétée en précisant que cette expérience n'est vraiment acquise que si elle est vécue.

Bien que la littérature sur la gestion de projet soit riche, et que les retours d'expérience sur des projets menés au sein de grands groupes ou lors de stages soient nombreux sur internet, ces derniers n'offrent qu'un cadre théorique au domaine du possible. En effet, chaque projet est différent du fait du contexte et des intervenants et de ce fait chaque retour d'expérience est différent d'un projet à l'autre et d'une personne à l'autre.

Ce que j'ai pu remarquer lors de la conduite de ce projet c'est, entre autres, la difficulté d'établir un planning et d'évaluer les délais des différentes étapes lorsque l'on débute un nouveau projet sans qu'il n'y ait de référentiel précédent. Ce référentiel se constitue par l'expérience de projets similaires. Or, dans notre cas, nous n'en avions pas et je n'avais qu'une vision incomplète des délais nécessaires à l'accomplissement de certaines tâches. Le planning initial a donc été amené à évoluer au fil du temps et de la clarification de certaines étapes ; les délais ont été par là même réévalués. A l'issue de ce stage un nouveau planning prévisionnel a été élaboré tenant compte du travail effectué et du travail restant à faire. Ce planning comportait 143 jours hommes de temps de travail réparti en deux grands axes :

- 36 jours pour la migration de la version 3 à la version 4 du *CRM Microsoft.* Cet axe incluait la modification des applications tierces, pour les rendre compatibles avec la nouvelle version du CRM. Les phases de test nécessaires à la bonne intégration de cette version dans l'environnement de production en faisaient également partie.
- 107 jours pour la mise en place de la CRM commerciale. Cet axe comportait 80 jours de développement servant à incorporer les personnalisations du CRM, à adapter les applications tierces pour le changement de la CRM commerciale et optimiser le logiciel de reprise de données. 25 jours étaient consacrés à la phase de test et d'implémentation ainsi qu'à la rédaction de la documentation. Enfin les 2 jours restant servaient à la mise en place de la solution et à la formation des utilisateurs.

De par mon travail précédent j'avais déjà eu l'occasion de mener à bien des projets. Cependant ceux-ci n'étaient pas des projets de développement faisant intervenir un panel si différent d'interlocuteurs internes et nécessitant la conduite d'une analyse des processus. J'ai pu constater à cette occasion que, bien que cela soit parfois difficile, il était nécessaire d'impliquer et de faire adhérer les futurs utilisateurs au projet.

Une des difficultés dans un environnement de production est de faire expliquer aux utilisateurs leurs méthodes de travail surtout lorsqu'ils font ce travail depuis longtemps. Si l'on demande par exemple à quelqu'un d'écrire une procédure décrivant comment faire du café on s'apercevra que suivant la personne telle ou telle étape ne sera pas présente car elle lui paraîtra intuitive. Ceci est également vrai dans l'analyse car des processus qui peuvent paraître simples aux utilisateurs expérimentés peuvent cacher en réalité une certaine complexité. Ces processus peuvent d'autant plus être difficiles à décrire lorsque certaines des règles de gestion ont été établies par des personnes n'appartenant plus à la société et qu'elles sont donc incomprises ou obsolètes. Dans ce genre de cas il peut être intéressant d'impliquer la Direction ou une instance stratégique afin qu'elle puisse prendre des décisions sur le devenir de ces règles.

Une autre difficulté est d'arriver à intégrer l'existant. Il est parfois plus simple de faire une analyse en partant de zéro pour un nouveau logiciel plutôt que de se trouver dans un environnement où un logiciel existe déjà. En effet les utilisateurs auront tendance à se référer systématiquement à l'ancien système et compareront l'ancien au nouveau avec le risque de vouloir reprendre à l'identique, défauts inclus, l'ancien système dans le nouveau. La difficulté est d'autant plus grande lorsque l'existant est en place depuis longtemps, dans le cas présent une quinzaine d'année. Dans ce contexte les utilisateurs ont tendance à faire l'amalgame entre outil de travail et méthode de travail, rendant l'analyse d'autant plus compliquée à effectuer. Dans ce genre de configuration, il faut savoir s'adapter et adapter ses propres méthodes de travail aux utilisateurs afin de les accompagner dans l'analyse et l'expression de leurs besoins.

Lors de l'analyse de ce projet j'ai été confronté à ces différents cas de figure et certaines des solutions ne sont venues qu'à postériori. J'ai pu également constater, comme d'autres

avant moi [8], qu'il était difficile de trouver le juste équilibre entre technique et gestion de projet lorsque l'on a la double fonction de responsable de projet et d'analyste programmeur.

Malgré ces difficultés je considère que faire participer les futurs utilisateurs au processus d'analyse est une nécessité. Cependant, il faut que chacun des participants comprenne les tenants et les aboutissants de l'analyse et puisse voir la progression du travail effectué. Afin d'en arriver là les utilisateurs doivent être formés aux méthodes d'analyse pour qu'ils en comprennent l'intérêt. En complément, des prototypes ou maquettes doivent être utilisés pour leur faciliter la vision du travail effectué.

J'ai pu constater également que le travail en groupe restreint, de l'ordre de deux ou trois personnes maximum, voir au travers d'entretiens individuels, permettait une meilleure synergie en laissant plus de temps de parole aux différents interlocuteurs et en leur permettant une meilleure implication. Pour la suite du projet nous avons donc décidé de travailler avec des groupes restreints d'utilisateurs et de recourir à des réunions plénières du groupe de travail pour confronter les différents points de vue et présenter le résultat des réflexions.

Chapitre 5 Prolongement du projet

A la fin de ce stage ce projet va continuer à évoluer au sein du groupe pour une mise en place effective au 31 décembre 2012. Mr LEDIEU, en charge du service, reprendra la gestion et l'analyse du projet pendant que je serai affecté à la partie technique de ce dernier. En ce qui concerne l'analyse, nous continuerons d'évoluer avec le groupe de travail mis en place au début de ce projet.

Malgré le travail effectué de nombreuses étapes restent encore à réaliser pour mener à terme ce projet. Nous devrons également faire face à des problèmes révélés par les tests de reprise de données effectués sur un échantillonnage.

En effet, les temps de traitement constatés, extrapolés à l'ensemble des enregistrements, donnent des temps de l'ordre de plusieurs jours, ce qui n'est pas acceptable. Plusieurs pistes existent pour optimiser ces temps de traitement. La première d'entre elles est de vérifier que le code du traitement est bien optimisé en limitant les boucles, les accès aux bases de données et les chargements d'objets au strict minimum. La deuxième est de modifier l'approche du traitement. Cela signifie qu'en considérant que le code du traitement est optimisé, il faut envisager de pouvoir effectuer l'importation des données par tranche d'enregistrements sur plusieurs jours. Il faut donc prendre en considération les données modifiées, supprimées et créées sur ce temps d'importation. Une possibilité consiste à transformer l'interface Web en application exécutable autonome travaillant sur un nombre d'enregistrements limités. De cette manière l'application peut être planifiée pour un traitement en tâche de fond pendant que les utilisateurs continuent de travailler sur l'ancien système. Une optimisation de ce système pourrait consister en la mise en parallèle des traitements d'importations.

Lors de l'importation, des anomalies ont été constatées dans la base de données source. Un travail de consolidation et de correction de ces anomalies devra donc être effectué par le service commercial dans la base de l'application *Selligent*.

Les tests d'intégration nécessitaient de remonter fréquemment des captures instantanées du serveur de base de donnée. Il s'est avéré que cela pouvait être pénalisant

pour l'équipe de développement travaillant sur le logiciel *Microsoft CRM* 3. De plus, la plateforme de production ayant évolué depuis le début du projet, il serait intéressant pour la suite de refaire la plateforme de développement en constituant deux réseaux de test indépendants. L'un servirait pour les tests de migration vers le logiciel *Microsoft CRM* 4 de Microsoft et l'intégration des développements liés à ce changement, l'autre servirait pour la maintenance de la plateforme de production. Cette configuration permettrait une meilleure flexibilité. Ces deux plateformes permettraient également de tester les modifications apportées aux applications périphériques en vue, dans un premier temps, du changement de version du logiciel *Microsoft CRM*, puis dans un deuxième temps, du changement de solution CRM commerciale.

Le travail de consolidation et d'optimisation du code des personnalisations du logiciel de CRM doit être poursuivi afin d'améliorer la réactivité de l'application et d'en simplifier la maintenance. Ce travail permettra par la suite de simplifier l'utilisation et l'intégration des nouveaux services Web fournis par Microsoft en vue de rendre les personnalisations compatibles avec les futures versions du logiciel de CRM de Microsoft.

Enfin, et surtout, l'analyse devra être poursuivie en collaboration avec le groupe de travail afin de finir d'identifier les données devant être récupérées et au final incorporer les automatismes liés au processus commercial.

Conclusion

Lors de ce mémoire j'ai été amené, en collaboration avec un groupe de travail, à effectuer une analyse des processus commerciaux ayant cours au sein de l'entreprise. Etant donné qu'aucune méthode d'analyse n'était définie, ou ne faisait consensus au niveau de l'entreprise, j'ai dû effectuer une recherche d'outils d'analyse ; cette recherche ayant abouti sur le choix de la méthode BPM. Les membres du groupe de travail, et moi-même, avons donc été sensibilisés à cette méthode qui nous a permis de formaliser un certain nombre de processus. Celle-ci sera sans doute réutilisée dans la suite du projet et je l'emploierai dans l'avenir pour la formalisation de processus sur d'autres projets.

J'ai également été amené à étudier les applications du système d'information et leurs interconnections. Grâce à cette étude j'ai pu être force de proposition sur des améliorations de ce système en le projetant sur de nouvelles technologies, le rendant ainsi plus pérenne et plus apte à évoluer. D'autre part, au travers de cette étude et de la création d'un logiciel de reprise de donnée, j'ai pu me familiariser avec différents langages et concepts de programmation. Ces concepts me seront utiles dans la suite de ma carrière.

Le sujet de ce mémoire peut paraître étrange de prime abord au vu de ma spécialisation en réseau système et multimédia. Il m'a cependant apporté des outils, des méthodes et une expérience qui me servirons dans ma carrière d'ingénieur tels que la conduite de projet, le travail en groupe ou encore les relations avec les utilisateurs. Ce sujet m'a également permis de mettre l'accent sur la partie multimédia, au sens large du terme, du cursus IRSM tout en requérant des compétences vues au fil de ma formation.

Perspectives

Le travail effectué représente les premiers fondements d'un projet global. La migration d'une solution de CRM vers une autre est la première étape permettant d'effectuer une consolidation des informations et des outils de la société.

Une fois ce travail terminé, une réflexion visant à mettre en place une méthodologie de CRM pourra être effectuée. Cette réflexion permettra de simplifier et d'harmoniser les processus commerciaux au niveau du groupe. Les différents outils en place aujourd'hui pourront alors être consolidés au sein de l'application de CRM permettant ainsi une meilleure réactivité et compréhension des clients.

En effet, la consolidation du fichier clients au sein d'une seule et même application offre la possibilité d'un couplage informatique-téléphonie (CTI) global permettant une meilleure prise en charge des clients. Elle permet de limiter le nombre d'applications utilisées par les collaborateurs, simplifiant ainsi leur travail en leur permettant de se concentrer sur leurs tâches principales. De plus, l'intégration des données commerciales au sein des données du service hotline permettra une meilleure compréhension des clients en facilitant le croisement des données au travers de rapports d'analyse.

Annexe 1 Business Process Modeling and Notation (BPMN)

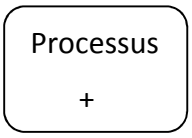
BPMN est un langage symbolique permettant de représenter des processus métier complexes sous forme de schéma. Il a été utilisé dans le cadre du projet afin de modéliser les processus métier inhérent à la gestion commerciale des clients. Afin de mieux comprendre ce langage vous trouverez dans ce document les symboles de base utilisés ainsi que leur signification.

La méthode BPMN pouvant incorporer de nombreux symboles tous n'ont pas été décrits dans ce document. Les sites indiqués dans la partie référence situé à la fin de ce document permettent d'obtenir de plus amples informations.

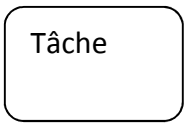
1 Symbolique de base du langage BPMN

1.1 Processus et tâches

Une activité est un travail effectué au court d'un processus industriel. Une activité peut être détaillée ou non. Les types d'activités dans un schéma de processus industriel sont au nombre de trois : les processus, les sous processus et les tâches.



Un processus (ou sous processus) est un « ensemble d'activités organisées dans le temps produisant un résultat précis et mesurable » (petit Robert) dont les entrées et les sorties sont clairement identifiables. Un processus est donc composé de tâches ou de sous-processus s'enchaînant de manière logique. Par exemple « démarrer une voiture » peut-être vu comme un sous processus du processus « utiliser une voiture ».

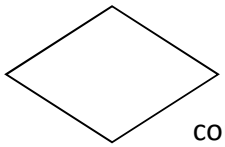


Une tâche est une étape identifiable du processus aux entrées et sorties clairement définies et dont la valeur ajoutée est mesurable. Il s'agit de l'élément de base d'un processus. Par exemple « tourner la clef » est une tâche du processus « démarrer une voiture ».

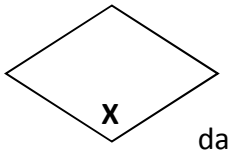
1.2 Embranchements conditionnels

Un embranchement conditionnel peut avoir une ou plusieurs activités entrantes et une ou plusieurs activités sortantes en fonction de la condition.

Il existe différents types d'embranchements conditionnels. Suivant la condition exprimée, celle-ci peut être appliquée aux activités entrantes ou sortantes.

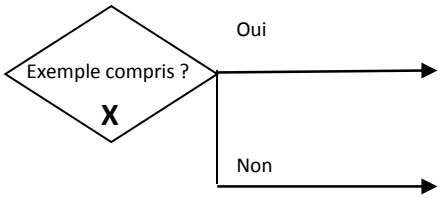


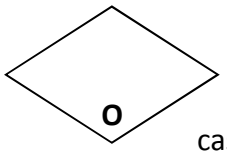
Ce symbole est la représentation générique d'un embranchement conditionnel.



Embranchement conditionnel de type « Ou Exclusif ». Cela signifie que dans le cas d'une condition s'appliquant en sortie sur deux branches l'une **ou** l'autre des branches sera prise **mais pas les deux**.

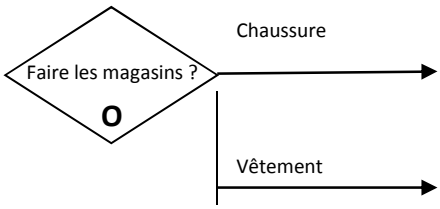
Exemple :

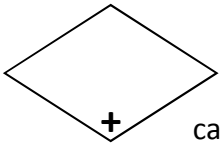




Embranchement conditionnel de type « Ou». Cela signifie que dans le cas d'une condition s'appliquant en sortie sur deux branches l'une **ou** l'autre des branches sera prise **voire les deux**.

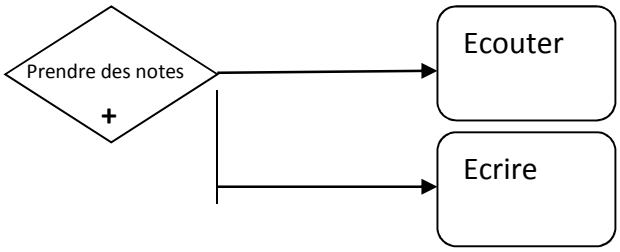
Exemple :

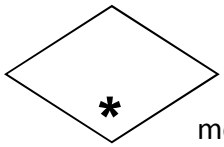




Embranchement conditionnel de type « Et». Cela signifie que dans le cas d'une condition s'appliquant en sortie sur deux branches **les deux seront prises**.

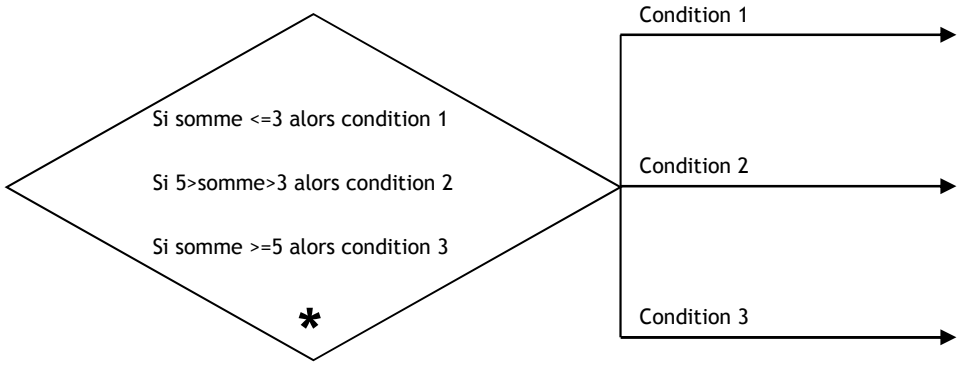
Exemple :





Embranchement conditionnel de type « Complexe». Dans ce cas le mode de fonctionnement de l'embranchement doit-être explicitement détaillé.

Exemple :



1.3 Etats

Les états des processus sont liés à des évènements extérieurs ou internes au processus qui vont influer sur ce dernier.

Un processus peut générer un évènement pour lui-même ou pour un autre processus.

Tout processus possède au minimum un état de début et un état de fin.



Etat de début représentant le point d'entrée du processus.



Etat intermédiaire.



Etat de fin de processus

Les états peuvent être liés aux évènements en précisant sous forme de texte l'évènement déclencheur, l'initiateur ou le destinataire de l'évènement. Des symboles peuvent être également utilisés pour préciser le support utilisé (par exemple, une enveloppe peut symboliser un e-mail ou un message).

Exemple :



Demande de documentation

(Client)

Ce schéma peut symboliser un état de début, pour un processus d'envoi de document, suite à une demande de documentation de la part d'un client, cette demande ayant été envoyée par e-mail.

1. <u>Liaisons</u>

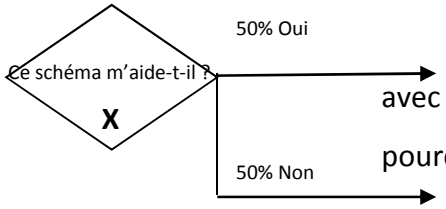


Les liaisons symbolisent les liens entre les activités. Elles sont orientées (flèche) afin de mettre en évidence les enchaînements entre les différentes activités.

Elles peuvent être complétées par des annotations permettant de les caractériser ou de les différencier dans le cas des embranchements.

Dans le cas des embranchements, des pourcentages peuvent également leur être associés afin de symboliser les chances de passage dans un des embranchements.

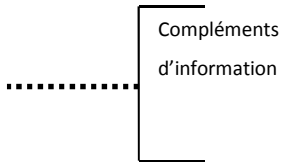
Exemple :



Embranchement conditionnel de type « Ou exclusif » avec deux branches (oui et non) ayant chacune le même pourcentage de chance d'être empruntée.

2. Annotations

Il est également possible de rajouter des annotations sur les activités ou les liaisons afin de rendre celles-ci plus claires.

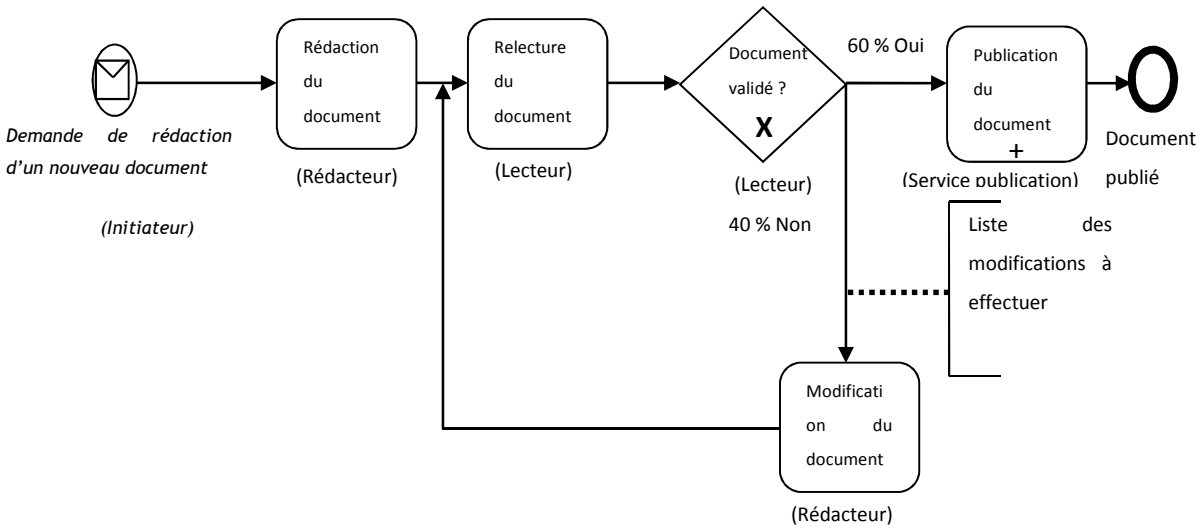


3. Exemple de schéma

Le schéma ci-dessous représente le processus de création et de publication de documents suivant :

- Un document est créé par un « Rédacteur » à la demande d'un « Initiateur », cette demande se fait sous forme de message (e-mail ou autre)
- Une fois le document rédigé, il est transmis pour validation à un « Lecteur ». Ce dernier émet alors un avis sur la validité du document.
- Dans 60 % des cas l'avis est favorable et le document est transmis à un service de publication pour le mettre à disposition, le sous-processus de publication n'étant pas détaillé.
- Dans 40 % des cas le document n'est pas validé. Le « Lecteur » doit alors transmettre au « Rédacteur » la liste des modifications à apporter.
- Une fois les modifications effectuées le document est soumis à nouveau à la relecture pour validation.

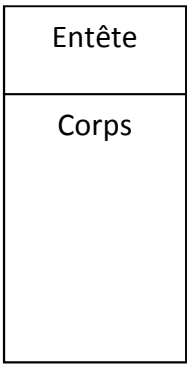
Remarque : Les méthodes de transfert d'information (e-mail ou autre) entre les différents acteurs n'ont pas été représentées pour simplifier l'exemple mais elles auraient pu être symbolisées sur les liaisons impactées ou sous forme d'« annotations » sur ces mêmes liaisons.



2 Symbolique avancée du langage BPMN

2.1 Pool

Un pool est utilisé dans un diagramme de flux, il permet de rendre de manière visuelle la répartition des activités en fonction des participants.

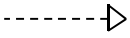


Un pool est représenté sous forme d'un rectangle horizontal ou vertical. Il est composé d'un entête, dans lequel est indiqué la liste des participants du pool, et d'un corps dans lequel sont représenté les activités associées à ce pool.

Les pools peuvent être encapsulés les uns dans les autres afin d'avoir un niveau de détail plus important

2.2 Flux

Les flux sont des liaisons entre des processus indépendant. Ils permettent de symboliser des échanges, ou des messages entre différent processus. Ils sont généralement utilisés pour symboliser des échanges entre des processus internes à une entreprise et des processus externes comme par exemple des échanges avec un client.



Un flux est représenté par une flèche vide en pointillé.

2.3 Autres symboles

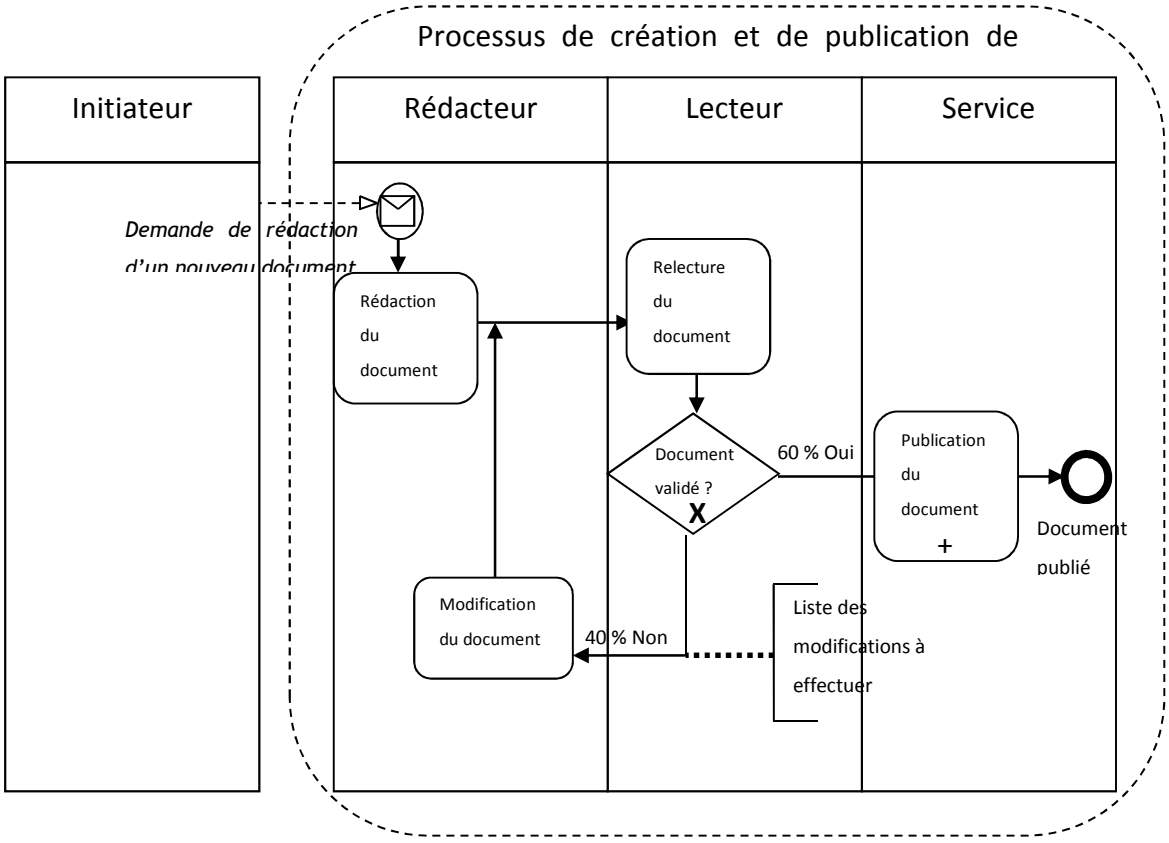


Il s'agit d'un évènement intermédiaire de type compteur de temps. Cela signifie que l'on attend un certain laps de temps avant de déclencher une activité. Ce symbole peut être directement rattaché à une activité.



Il s'agit d'un symbole rattaché à une activité. Il signifie que cette activité « tourne en boucle ». La boucle peut être rompue par un évènement extérieur, la fin d'un laps de temps défini ou du fait qu'un certain nombre d'occurrences de cette boucle se soit produites.

2.4 Exemple de schéma



Annexe 2 Compte-rendu de réunion GTP CRM Commercial

1 Synopsis

Participants présents :

Frédéric CHAPELON
Laurent FILOCHE
Laure VANDERLENNE
Pascal RAMDANI
Sandrine GUILLOT

Date de la réunion : 25/05/2010

Heure de début : 10h22

Heure de fin : 12h42

Points à l'ordre du jour :

- Approbation du le compte-rendu de la réunion du 26/04.
- Approbation du le compte-rendu de la réunion du 17/05.
- Report de la réunion du 07/06 au 08/06.
- Retour sur le schéma du processus d'avant-vente « Qualification de la demande »
- Retour sur les schémas du processus d'avant-vente « Traitement de la demande ».

Point reporté :

- Fin de l'étude du processus de pré-réservation

Résumé :

Le groupe de travail est revenu sur les schémas des processus d'avant-vente élaborés à partir des procédures transmises et a apporté les modifications nécessaires. Les schémas étudiés lors de cette réunion correspondaient aux processus de « Qualification de la demande », « Traitement de la demande », « Rendez-vous client », « Elaboration proposition commerciale », le processus « Pré-réservation » ayant commencé à être abordé. Les membres du groupe de travail ont demandé à ce qu'apparaissent de manière plus claire les liaisons entre les différents processus. Ils ont également émis le souhait d'avoir une

présentation du module de CRM technique et du module de CRM commercial lors d'une prochaine réunion.

2 Compte-rendu des points à l'ordre du jour

2.1 Approbation du compte-rendu de la réunion du 26/04

Pas de remarques particulières concernant ce compte rendu.

2.2 Approbation du compte-rendu de la réunion du 17/05

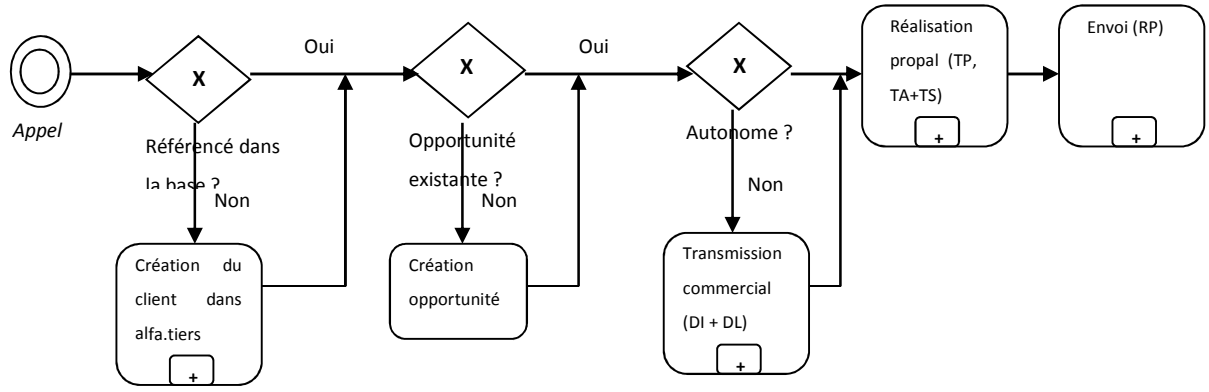
Pas de remarques particulières concernant ce compte rendu.

2.3 Report de la réunion du 07/06 au 08/06

La date du 08/06 a été retenue pour cette réunion.

2.4 Processus de « Qualification de la demande »

2.4.1 Schéma initial proposé à l'issue de la réunion du 17/05



Ce schéma a été modifié à la suite de la réunion du 17/05 afin d'intégrer certaines remarques issues de cette même réunion et de conserver un certain niveau d'information. A savoir :

« Réalisation propal (TP, TA+TS) » et « Envoi (RP) » sont plutôt des processus de traitement de la demande ils doivent donc être intégrés dans le processus correspondant.

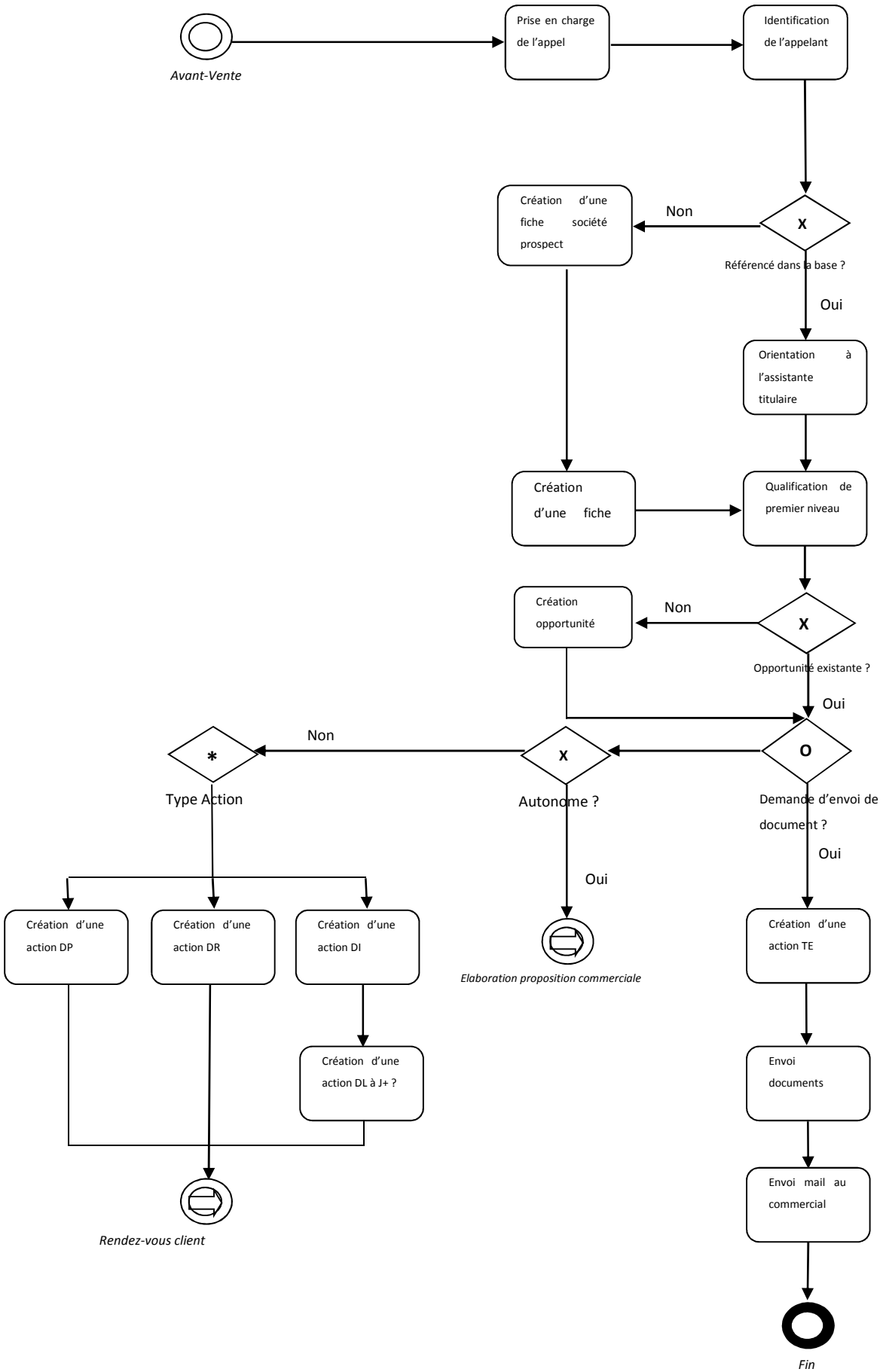
« Création du client dans alfa.tiers » a été scindé en « création de la fiche société » et « création de la fiche personne ».

« Transmission commercial (DI+DL) » a été détaillée et la création des actions DR et DP conservée.

Un embranchement supplémentaire représentant la demande de document de la part du client a été incorporé.

Un nouveau schéma incorporant ces modifications a donc été soumis au groupe de travail.

2.4.2 Schéma soumis à validation



2.4.3 Retour sur les points du schéma ayant été soumis à discussion ou précisés

a. Prise en charge de l'appel

Dans le cadre d'une généralisation du processus de « Qualification de la demande » cette tâche peut-être renommée « Prise en charge de la demande »

b. Création d'une opportunité

L'opportunité créée est de type commercial

c. Opportunité existante ?

Il est curieux que le traitement soit le même si l'opportunité existe ou non, une différenciation du traitement pourrait-être envisagée.

d. Demande d'envoi de document ?

Cette partie ainsi que les tâches découlant d'un envoi de documents apparaissent trop au niveau central, ce qui leur donne plus d'importance qu'elles n'en ont. De plus, il s'agirait plutôt d'un connecteur de type « ou exclusif ».

e. Création d'une action TE

Les actions de transmission externe (TE) ont un ciblage assez large (client, fournisseur, prestataire, etc.). Généralement, lors d'envois de documents au client, on utilise plutôt une action transmission courrier (TC) qui est plus orienté client. Dans le cas d'un envoi courrier ou d'un fax, le fichier envoyé est inséré dans l'action, dans le cas d'un mail le mail et la pièce jointe sont inséré dans l'action. Il peut être envisagé de créer une action spécifique à l'envoi de documents clients ou de récupérer les actions TC en les renommant transmission client ou transmission information client.

f. Envoi mail au commercial

L'envoi d'un mail au commercial suite à une demande de documents n'est pas une fin en soi. Le commercial, au bout d'un délai de 2 semaines, devrait contacter le client et voir avec lui s'il est intéressé par nos produits suite à la réception de la documentation. Dans le cas où il est intéressé, le commercial lui propose un rendez-vous (PR). Sinon le processus se termine.

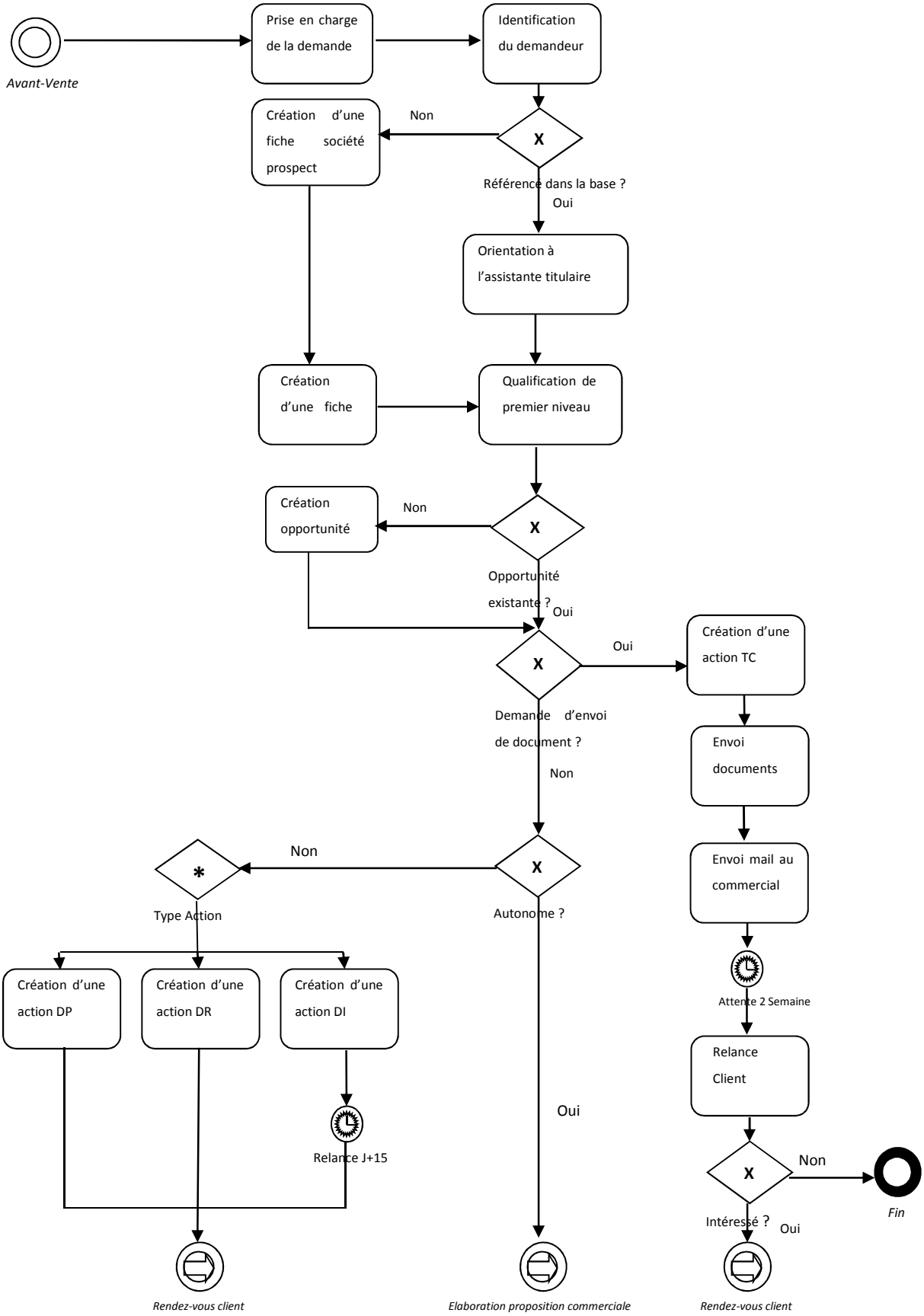
g. Création d'une action DP

Une action de demande d'information complémentaire (DP) ne débouche pas systématiquement sur une prise de rendez-vous. Elle peut, dans certains cas, déboucher éventuellement sur une transmission de documents.

h. Création d'une action DL

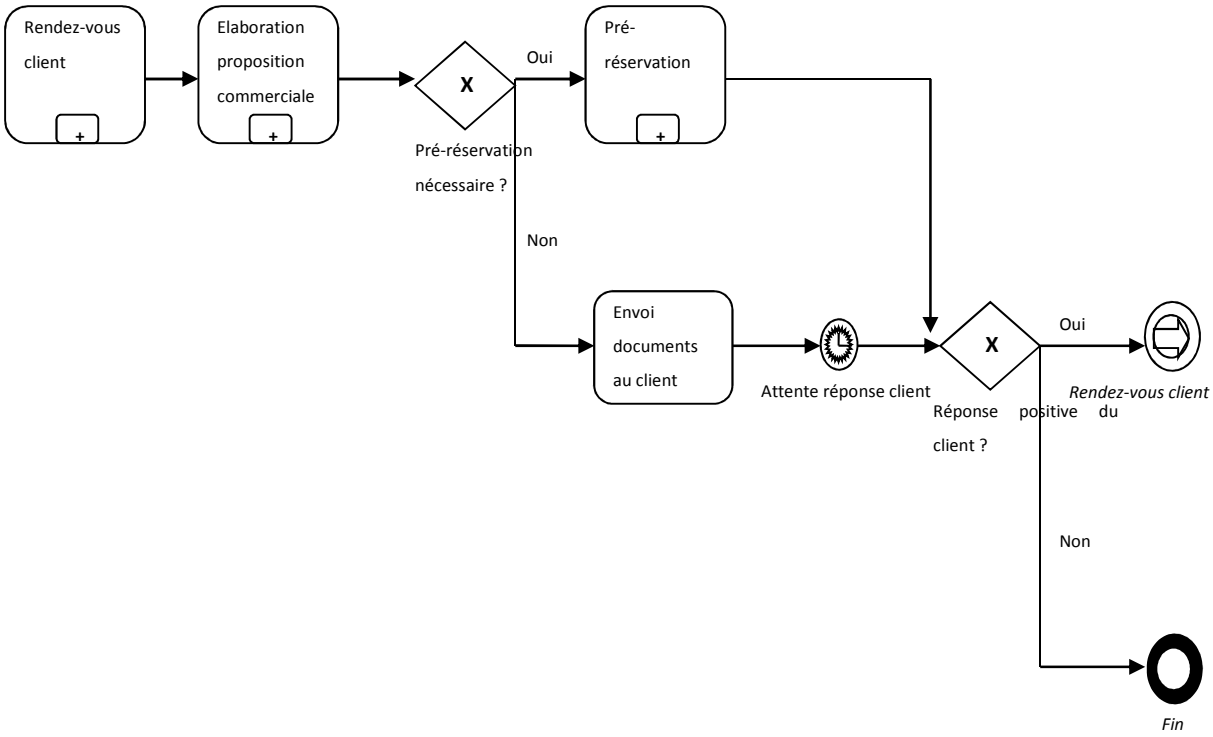
L’action DL n’as pas forcément lieu d’être ici. Il s’agit plus d’un système de relance du commercial après un délai de 15 jours. On peut envisager un système automatique de relance ou même la création d’une action particulière qui ne serait pas noyée dans les actions DL du commercial.

2.4.4 Schéma modifié



2.5 Processus de « Traitement de la demande »

2.5.1 Schéma initial



2.5.2 Retour sur les points du schéma ayant été soumis à discussion ou précisés

a. Pré-réservation

Le processus de pré-réservation doit déboucher sur la tâche « envoi documents au client ».

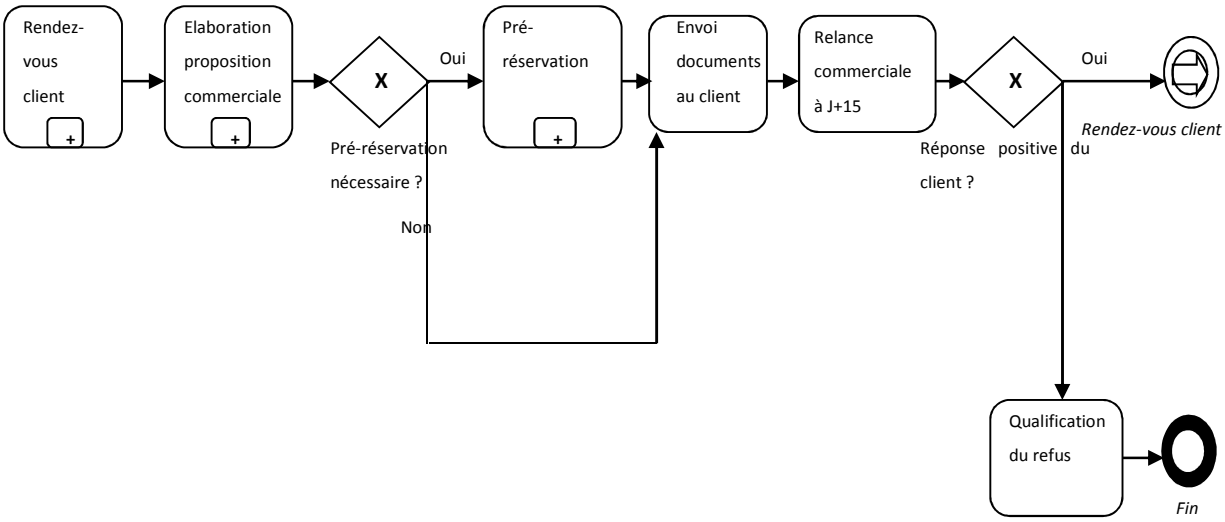
b. Attente réponse client

Nous n'attendons pas la réponse du client, une tâche de relance (action RP) est affectée au commercial pour qu'il recontacte le client à J+15 après l'envoi du document.

c. Réponse positive du client ?

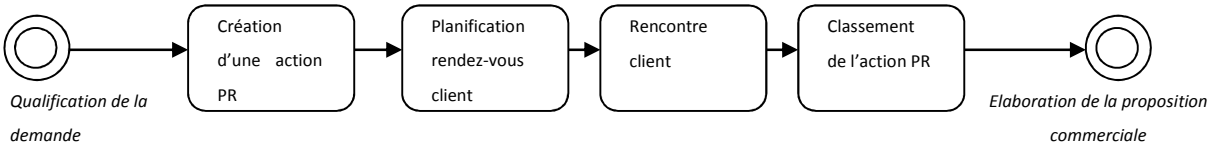
Lorsque le client refuse la proposition par une réponse négative l'opportunité est clôturée et une qualification du refus doit être effectuée. Dans le cas où l'opportunité est perdue à cause d'un concurrent ce dernier doit être identifié.

2.5.3 Schéma modifié



2.6 Processus « Rendez-vous client »

2.6.1 Schéma initial

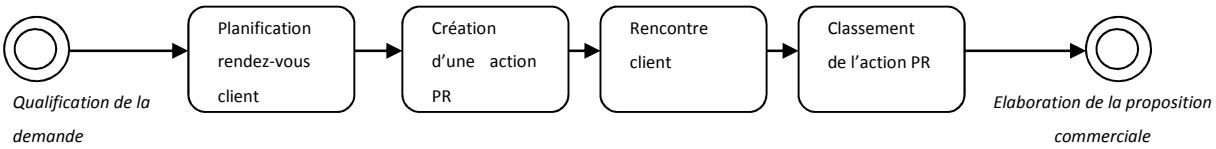


2.6.2 Retour sur les points du schéma ayant été soumis à discussion ou précisés

a. Création d'une action PR

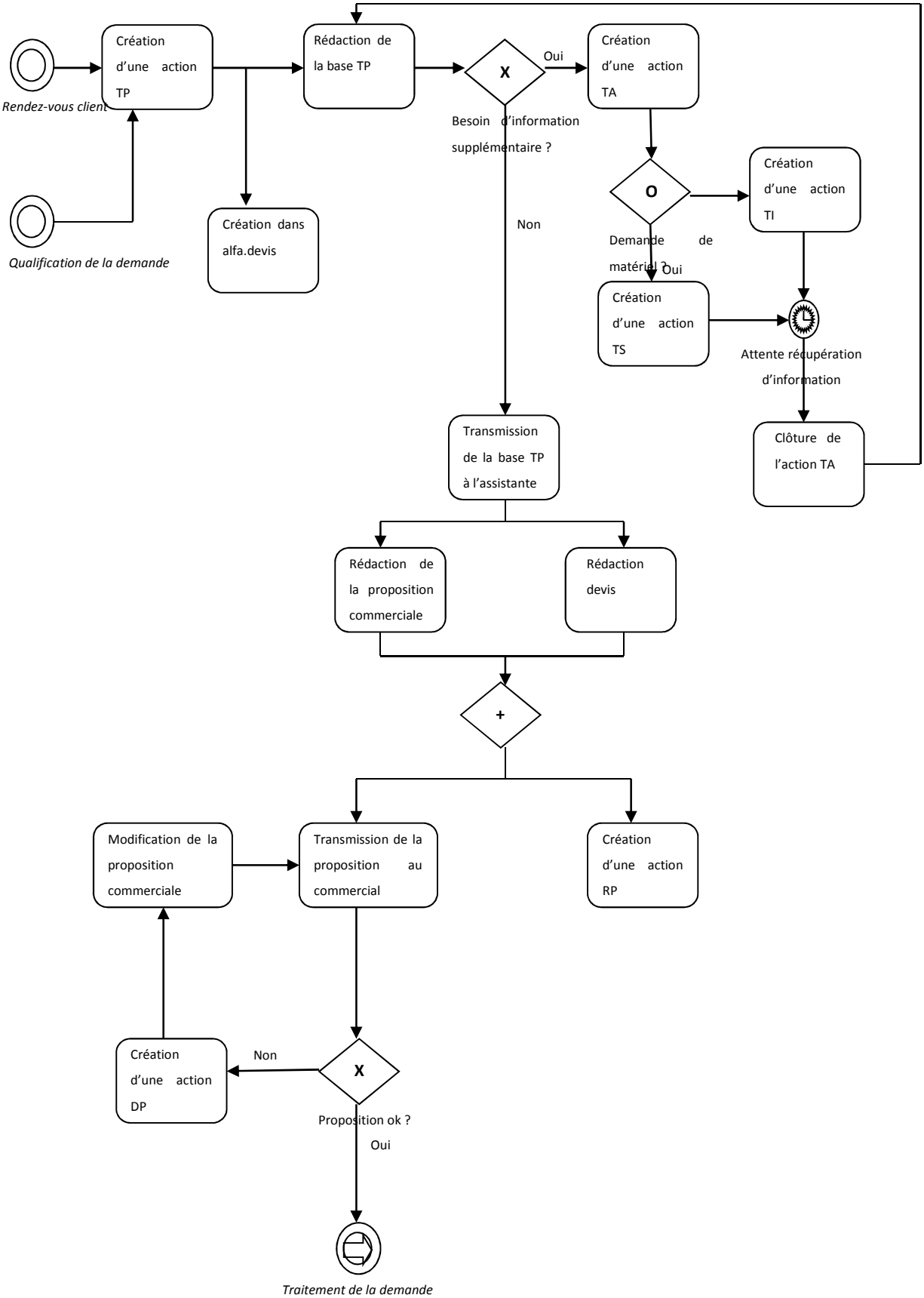
La création de l'action PR a lieu après avoir contacté le client pour planifier le rendez-vous.

2.6.3 Schéma modifié



2.7 Processus « Elaboration de la proposition commerciale »

2.7.1 Schéma initial



2.7.2 Retour sur les points du schéma ayant été soumis à discussion ou précisés

a. Création d'une action TP

En termes d'évolution, l'action TP devrait être remplacée par l'opportunité commerciale. Cependant, se pose la question du stockage des versions des opportunités. Aujourd'hui les différentes versions d'une proposition sont représentées par différentes action TP. Cependant, il est mal aisé de suivre ces évolutions car chacune de ces actions ne contient que la modification et ne reprend pas l'historique des modifications précédentes. Il faut donc réfléchir à la possibilité de présenter une vue globale de la proposition tout en gardant l'historique des modifications de la proposition. Une solution proposée serait de faire une duplication de l'opportunité contenant la base. Une autre piste serait de voir ce qui est fait dans alfa.devis sur le stockage des versions des devis et voir si cela est applicable dans notre cas de figure. Il a été demandé aux membres du GTP de réfléchir à cette problématique afin que nous ayons une concertation sur la marche à suivre.

b. Création d'une action TA

En termes d'évolution, l'action TA devrait disparaître et être remplacée par un état « En attente » de l'opportunité. La création de cette action est faite par l'assistante commerciale après la réception de la base de la proposition.

c. Demande matériel ?

Ce connecteur n'a pas lieu d'être à ce niveau, la décision de créer une action de type TS se faisant dans alfa.devis.

d. Création d'une action TS

L'action TS est créée automatiquement par alfa.devis

e. Rédaction proposition commerciale

La rédaction de la proposition commerciale se fait à partir d'alfa.devis.

f. Rédaction devis

La rédaction du devis se fait à partir d'alfa.devis.

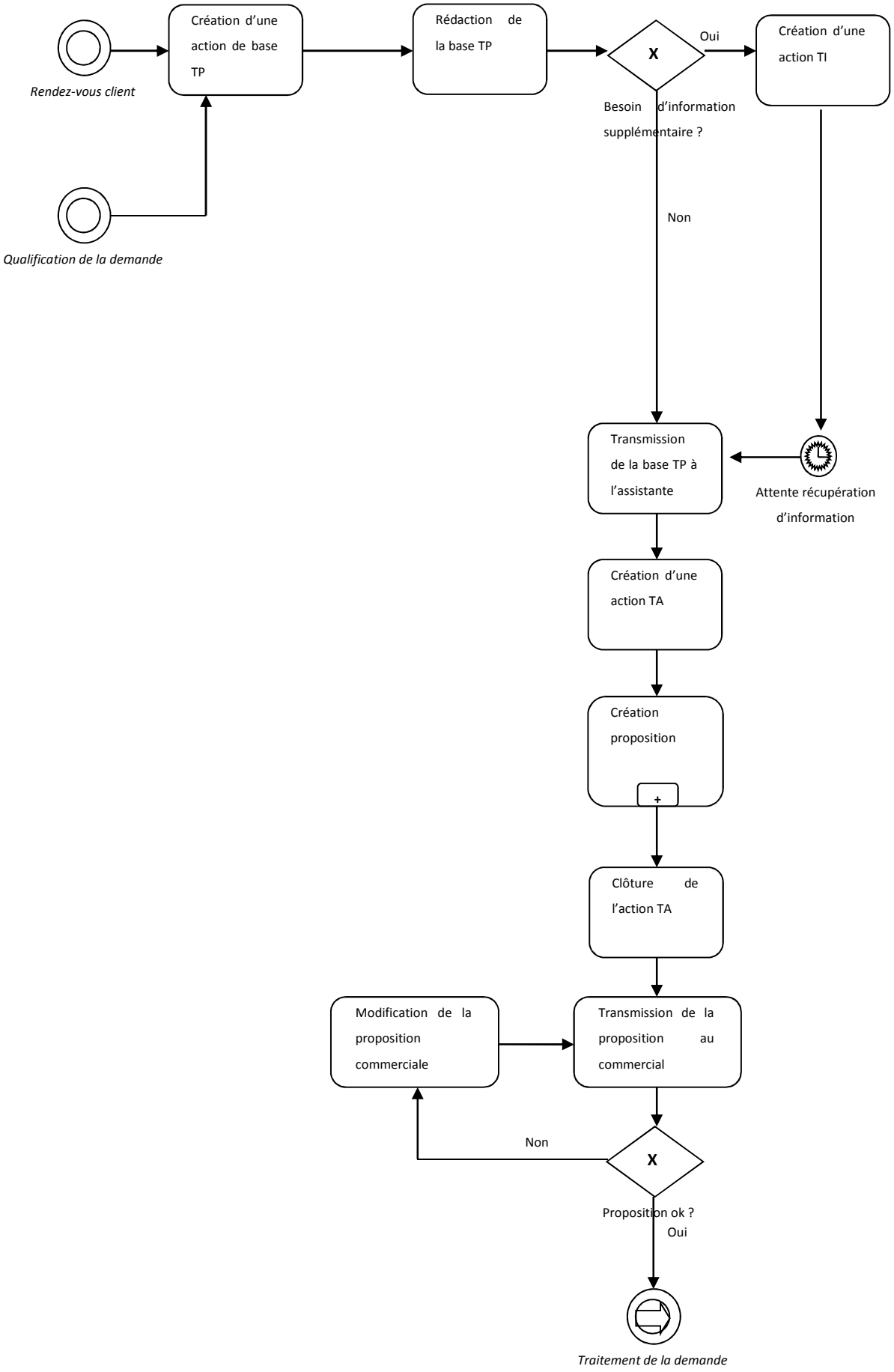
g. Création d'une action RP

La création de l'action RP est faite automatiquement à partir d'alfa.devis.

h. Création d'une action DP

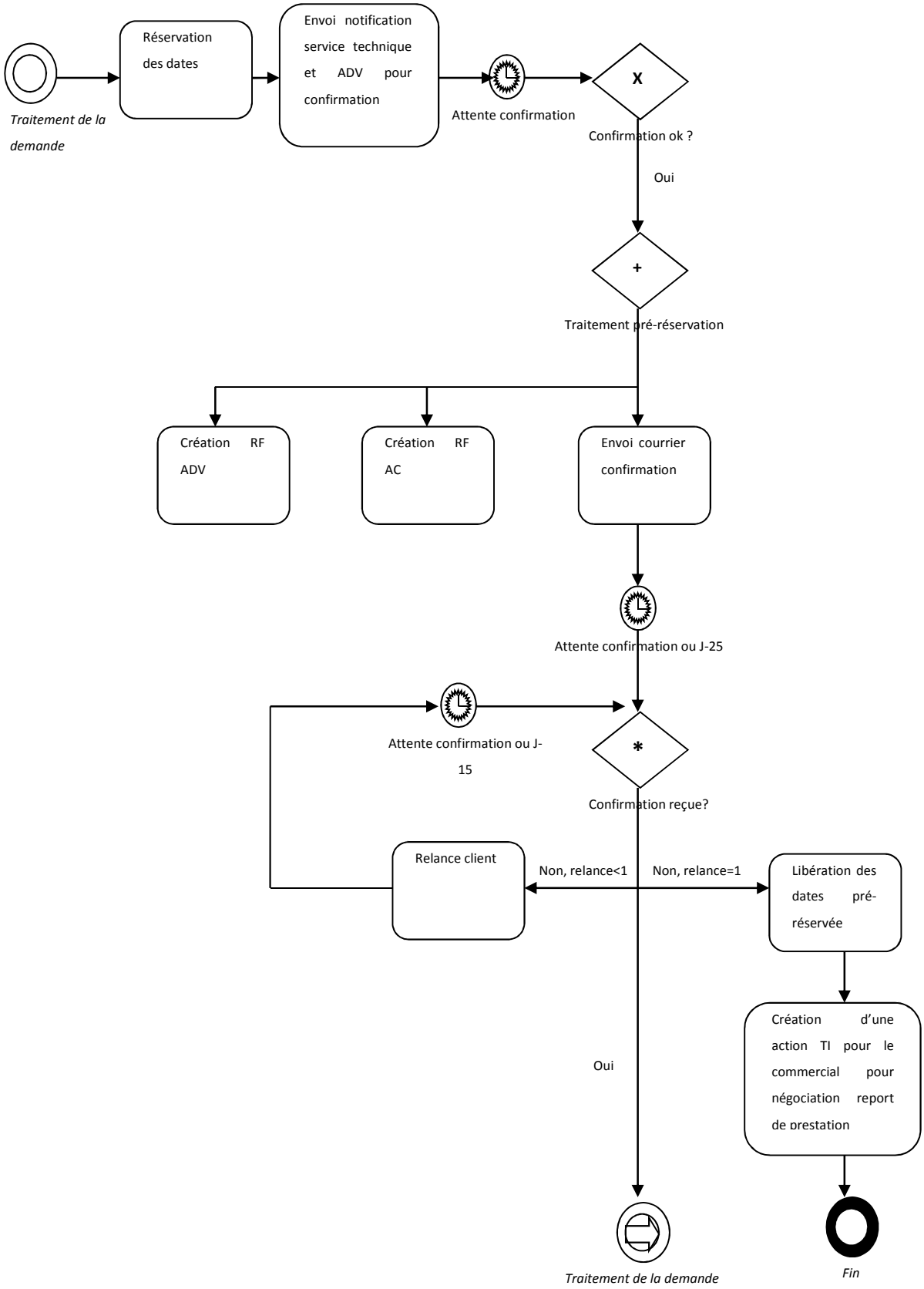
En cas de modification il n'y a pas d'action DP créé.

2.7.3 Schéma modifié



2.8 Processus « Pré-réservation »

2.8.1 Schéma initial



2.8.2 Retour sur les points du schéma ayant été soumis à discussion ou précisés

a. Pré-réservation

Ce processus à l'origine exceptionnel devient de plus en plus fréquent. Il peut être amené à être appelé de manière « indépendante ». Par exemple : un client ayant déjà reçu un devis peut conditionner la signature du devis à la communication préalable de date d'exécution de la commande. Ce processus est effectué par l'administration des ventes

b. Envoi notification service technique et ADV pour confirmation

Cette étape n'est pas effectuée aujourd'hui. Elle n'a pas de sens pour l'administration des ventes car c'est ce même service qui effectue le processus de pré-réservation. Cependant la notification au service technique peut être envisagée. Si cette étape se fait elle serait plutôt située en parallèle des créations des actions RF AC.

c. Création RF AC

Ces actions ne sont créées que dans le cadre d'un projet, en parallèle de l'action RP du commercial afin de pouvoir relancer ce dernier. Il s'agit d'une forme de relance de l'action RP du commercial.

d. Création RF ADV

Ces actions ne sont créées que dans le cadre d'une réservation simple, en parallèle de l'action RP du commercial afin de pouvoir relancer ce dernier. Il s'agit d'une forme de relance de l'action RP du commercial.

e. Envoi courrier confirmation

Il n'y a pas d'envoi de courrier de confirmation. Il peut arriver qu'il y ait un envoi de courrier contenant la proposition de dates mais il est plus fréquent que ce soit le commercial qui voie directement avec le client.

3 Compte-rendu des points n'étant pas à l'ordre du jour

3.1 Remarque sur les schémas

Il a été dit que les schémas en version imprimables étaient difficiles à lire, notamment sur le suivi des liaisons entre les différents processus. Il a donc été suggéré de mettre en place un système de numérotation sur les entrées et les sorties des processus pour rendre plus aisé le cheminement entre les différents schémas.

3.2 Présentation CRM

Les membres du GTP ont émis le souhait d'avoir une présentation du module de CRM technique afin de voir les possibilités offertes par le CRM, ainsi qu'une présentation du module de CRM commercial. Ils ont été informés que le module de CRM technique était déjà personnalisé pour le service client et qu'aucun développement n'avait été commencé sur le module de CRM commercial. Il a été convenu de procéder tout de même à cette présentation lors de la réunion du 8/06.

Annexe 3 Concepts commerciaux

Le but de cette annexe est de faire une présentation générale du modèle commercial de la société et du logiciel *Selligent* pour illustrer les propos du présent mémoire. Elle se veut volontairement simple pour des raisons de clarté afin de ne pas surcharger le lecteur d'information.

1 Modèle commercial de base

Le marché sur lequel se positionne la société ALFA Informatique implique certains concepts spécifiques. En effet les cibles de ce marché sont les associations de type loi 1901. Or, ces associations locales se regroupent parfois, au niveau légal, au sein d'entités juridiques. Du fait de ces regroupements, certaines personnes peuvent apparaître au sein de plusieurs établissements avec des fonctions différentes.

Pour mieux comprendre le modèle commercial de la société, voici quelques notions et définitions de base :

- Une entité juridique est composée de un ou plusieurs établissements.
- Un établissement ne dépend que d'une seule entité juridique.
- Une personne est liée au minimum à un établissement.
- Un établissement possède au moins un contact.
- Une opportunité est un regroupement d'actions ayant pour objectif d'aboutir à une vente. Elle est liée à un seul établissement et se compose de une à plusieurs actions. Un contact au sein de l'établissement est lié à l'opportunité.
- Une action correspond à toute action entreprise dans le but d'effectuer une vente. De ce fait, une action est toujours liée à une seule opportunité. Un contact de l'établissement est forcément lié à l'action. Cependant d'autres acteurs peuvent participer à l'action.
- Une campagne est un ensemble d'actions marketing visant à promouvoir un produit ou un ensemble de produits auprès d'une population cible. Du fait que les actions sont forcément liées à une opportunité, une campagne doit contenir au moins une opportunité liée à une action. Les actions de l'opportunité, quant à elles, ne sont pas forcément liée à la campagne.
- Un établissement peut être considéré comme suspect, prospect, client. Un suspect est un établissement que l'on suspecte d'être intéressé par un produit de la société mais

avec qui aucun contact n'a encore été établi. Un prospect est un suspect ayant été contacté, ou un client que l'on suspecte d'être intéressé par un autre produit de la société. Enfin un client est un prospect ayant signé une opportunité et donc ayant acheté un produit.

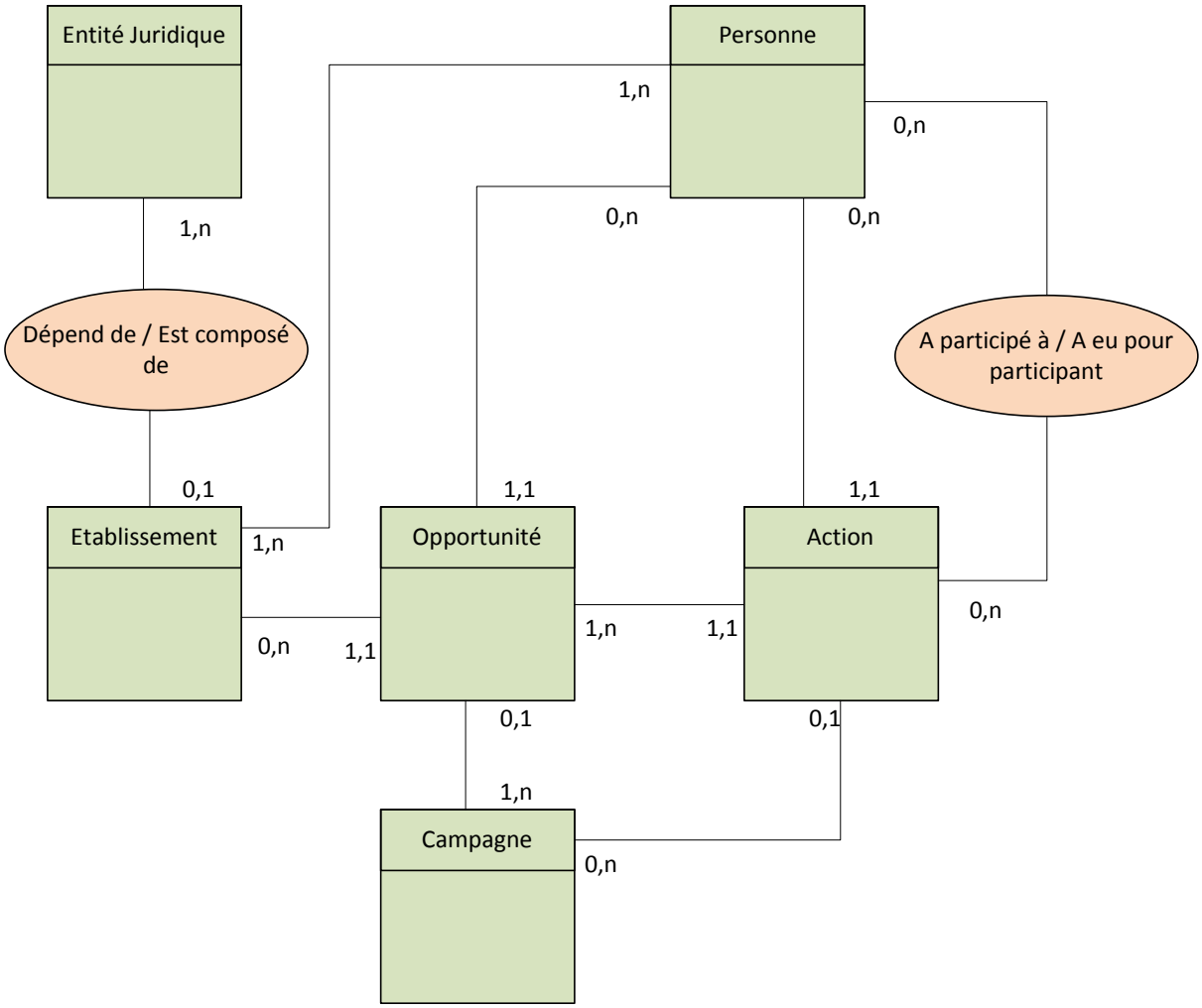


Figure 16 : Schéma des liens entre les concepts de base

2 Liens entre chapitres au sein de *Selligent*

Le logiciel *Selligent* organise les notions commerciales au sein de chapitres. Le modèle commercial de l'application est proche de celui de la société. Les cardinalités entre les différents chapitres sont légèrement différentes du modèle d'ALFA Informatique du fait que ces dernières sont liées à des règles de gestion. Cependant ces différences ont été en partie palliées par des scripts de contrôle mis en place dans l'application.

Le chapitre société a été utilisé pour regrouper différentes notions notamment celles d'entité juridique, d'établissement, de fournisseur, de partenaire et de concurrent. Il est possible de définir les liaisons, comme par exemple « Dépend de / Est composé de », entre les différents chapitres. Pour des raisons de clarté ces liaisons n'ont pas été représentées dans le schéma.

L'ensemble de ces adaptations permet donc d'être au plus proche du modèle commercial de la société.

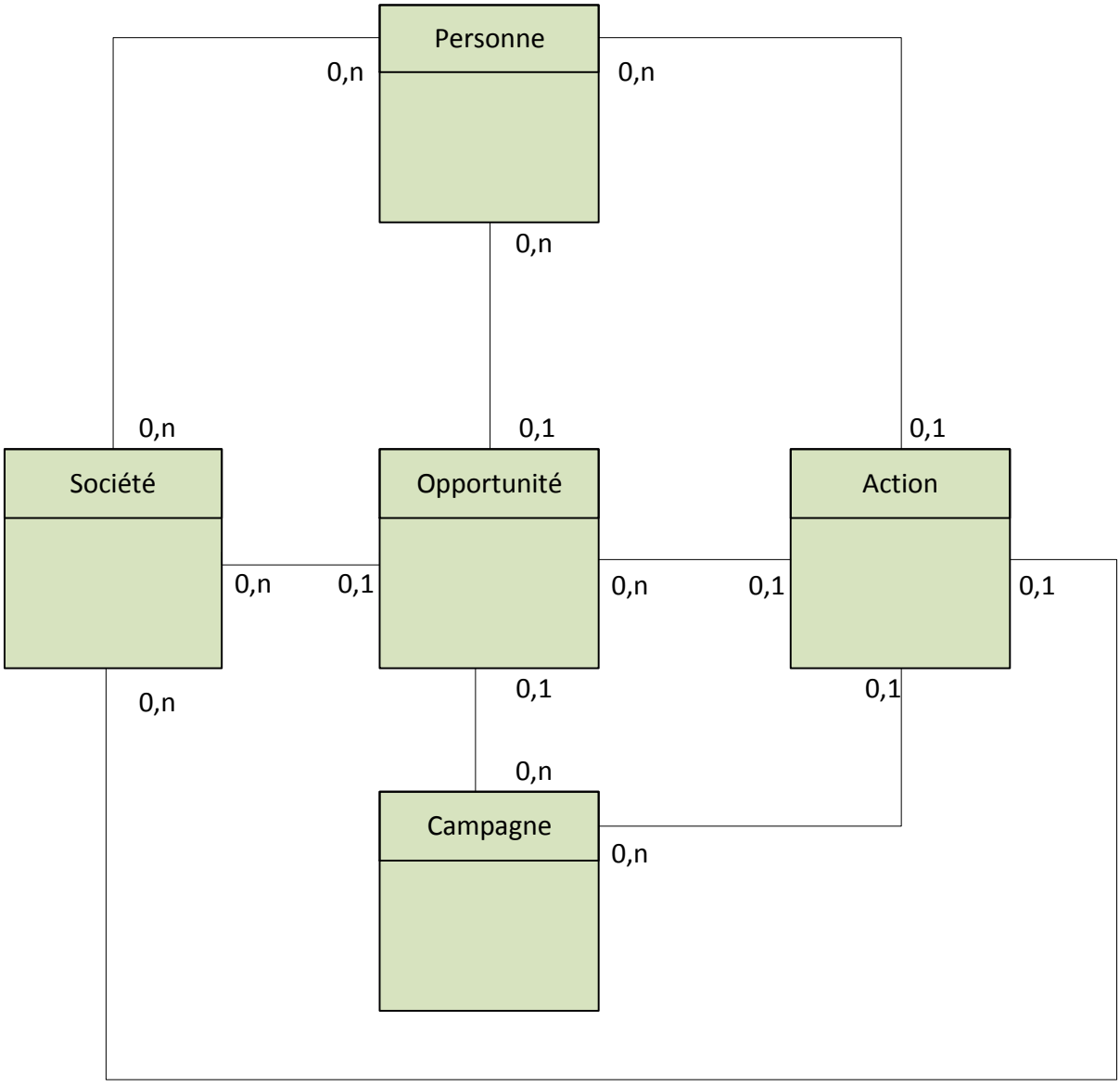


Figure 17 : Schéma des liens entre les chapitres principaux de *Selligent*

Annexe 4 Les différentes architectures d'applications

Une application peut être divisée en trois parties abstraites distinctes [9] [10] Ces trois parties pouvant être organisées en couche :

- Couche Présentation : cette couche peut également être appelée IHM (Interface Homme Machine). Comme son nom l'indique, cette couche est responsable de la représentation des données et de l'interaction avec l'utilisateur (clavier, souris).
- Couche Traitement : elle correspond à la mise en œuvre de l'ensemble des règles de gestion et de logique applicative. Elle peut être subdivisée en deux sous couches :
    - Traitement locaux : il s'agit des traitements effectués en relation avec l'IHM (exemple : un contrôle de saisie) ;
    - Traitement globaux : également appelée couche métier, elle a pour rôle de regrouper la logique applicative, c'est-à-dire les règles élémentaires et internes qui régissent les processus applicatifs.
- Couche Données : elle représente les mécanismes permettant de gérer les données destinées à être conservées sur la durée.

Ces couches sont soumises à des règles strictes d'ordonnancement, proche du modèle OSI (Open System Interconnexion) en réseau [11], faisant que chaque niveau ne communique qu'avec ses voisins, qu'un niveau de couche ne peut consommer des services que d'une couche de niveau inférieur et ne met à disposition ses services qu'à une couche de niveau supérieur [12].

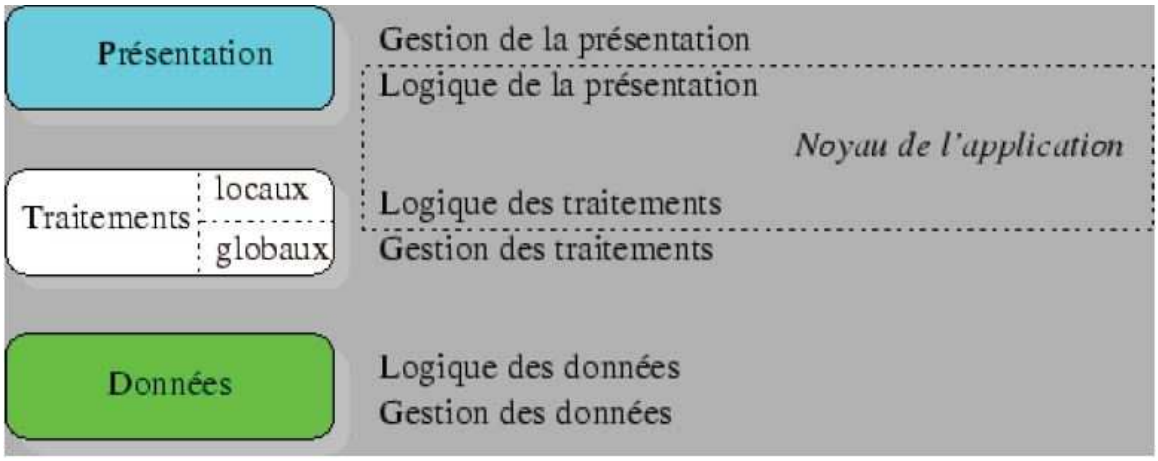


Figure 18 : Les trois niveaux d'une application informatique [9]

La répartition, ou l'imbrication, de ces différentes couches sur diverses machines permettent de définir quatre schémas d'architectures d'application distincts [9].

1 Les applications un-tiers

Historiquement, ces architectures applicatives ont été les premières utilisées. Les trois couches sont intimement liées et s'exécutent sur une même machine. Elles peuvent être soit utilisées sur site central (mainframe), soit déployées sur plusieurs machines indépendantes.

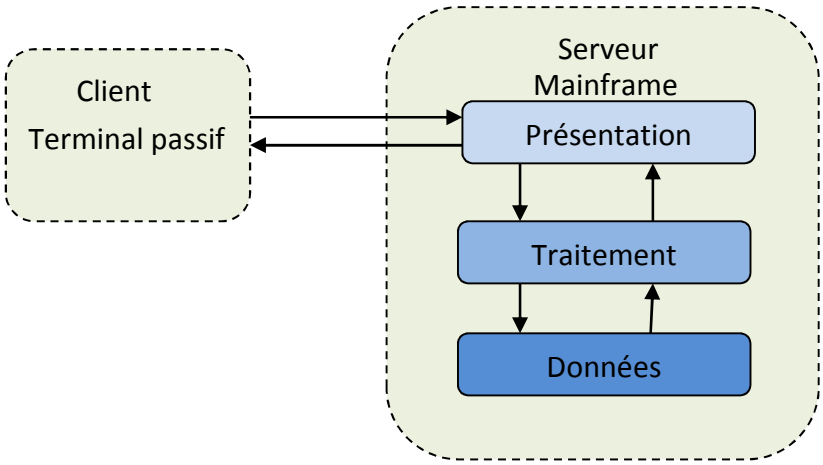


Figure 19 : Schéma d'une architecture Mainframe

Les solutions mainframe, bien que réputées fiables et offrant une maintenance facile du fait de leur centralisation, ont été fortement délaissées suite à l'émergence des micro-ordinateurs et de leur environnement bureautique. Elles ont en effet été pénalisées par leur faible capacité à interagir avec les applications bureautiques. De telles solutions restent aujourd'hui cantonnées à des niches applicatives spécifiques nécessitant des traitements par

lots importants, même si l'on constate un regain d'intérêt concernant la centralisation de la puissance de calcul avec l'émergence des solutions de virtualisation [13].

Les solutions déployées, quant à elles, bien que proposant une interface riche et pouvant interagir avec l'environnement bureautique de l'utilisateur, ont été délaissées dans le cas des environnements multiutilisateurs car elles posaient des problèmes d'accès aux données et d'intégrité des données. En effet, la cohabitation de plusieurs moteurs de base de données indépendants manipulant les mêmes données peut devenir assez instable et il n'est pas rare de rencontrer des conflits lors de la consultation ou de la modification simultanée d'un même enregistrement par plusieurs utilisateurs [10].

La volonté de tirer le meilleur parti des deux solutions, à savoir la stabilité, la fiabilité des solutions mainframe et la richesse de l'interface utilisateur des solutions déployées, a conduit à l'émergence des architectures deux-tiers.

2 Les applications deux-tiers

Dans une architecture deux-tiers, également appelée architecture client-serveur de première génération, le poste client se contente de déléguer la gestion des données à un service spécialisé.

Cette architecture permet d'utiliser la puissance des postes clients et de fournir une interface riche aux utilisateurs tout en garantissant la cohérence des données. Dans ce modèle, la communication entre le client et le serveur est effectuée au travers d'un intergiciel (middleware [9]) dont le rôle est de masquer la complexité du dialogue client-serveur. Au travers de ce middleware, le client va initier une communication avec le serveur de données, ce dernier se contentant de répondre aux demandes des clients.

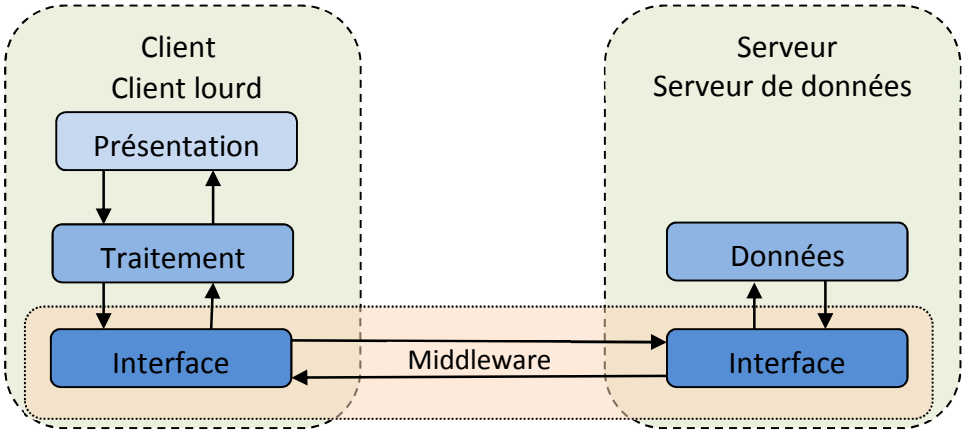


Figure 20 : Schéma d'une architecture client-serveur

Malgré le fait qu'il offre une interface riche, ayant permis l'appropriation des applications par les utilisateurs, ce modèle d'architecture connaît de nombreuses limites qui sont à l'origine de la notion de client lourd (par opposition au client léger des serveurs mainframe) :

- On ne peut soulager la charge du client qui supporte tous les traitements applicatifs.
- Les mises à jour applicatives sont plus difficiles à mettre en place.
- Les interactions fortes entre le client et le serveur rendent plus difficiles les modifications de structure de la partie serveur.
- Le dialogue client-serveur est assez conséquent, ce qui cantonne l'utilisation de ce type d'architecture aux réseaux locaux ou à large bande.

Les architectures distribuées du type trois-tiers cherchent à pallier ces inconvénients tout en conservant les avantages de l'architecture deux-tiers.

3 Les applications trois-tiers

Cette architecture est également appelée client-serveur de deuxième génération. Elle consiste en la séparation de l'application en trois niveaux [9] [10] [12] :

- Premier niveau : la couche présentation et les traitements locaux sont pris en charge par le client.
- Deuxième niveau : les traitements globaux sont pris en charge par un service applicatif.
- Troisième niveau : les données sont prises en charge par un serveur général de base de données (SGBD).

Ces différents niveaux étant indépendants les uns des autres, ils peuvent être répartis sur différentes machines physiques.

Généralement, dans cette architecture, le premier niveau sera présent sur le poste client. Ce dernier étant soulagé de la partie traitements globaux, il pourra être moins évolué. On qualifie habituellement ces postes de client léger par opposition au client lourd de

l'architecture deux-tiers, mais ils présentent plus de possibilités que les clients des architectures mainframe.

Le deuxième niveau est habituellement présent sur un serveur qualifié de serveur applicatif dont le rôle est de fournir des services qui seront consommés par le client. Ce serveur, quant à lui, accédera aux données en utilisant les processus de l'architecture deux tiers de première génération.

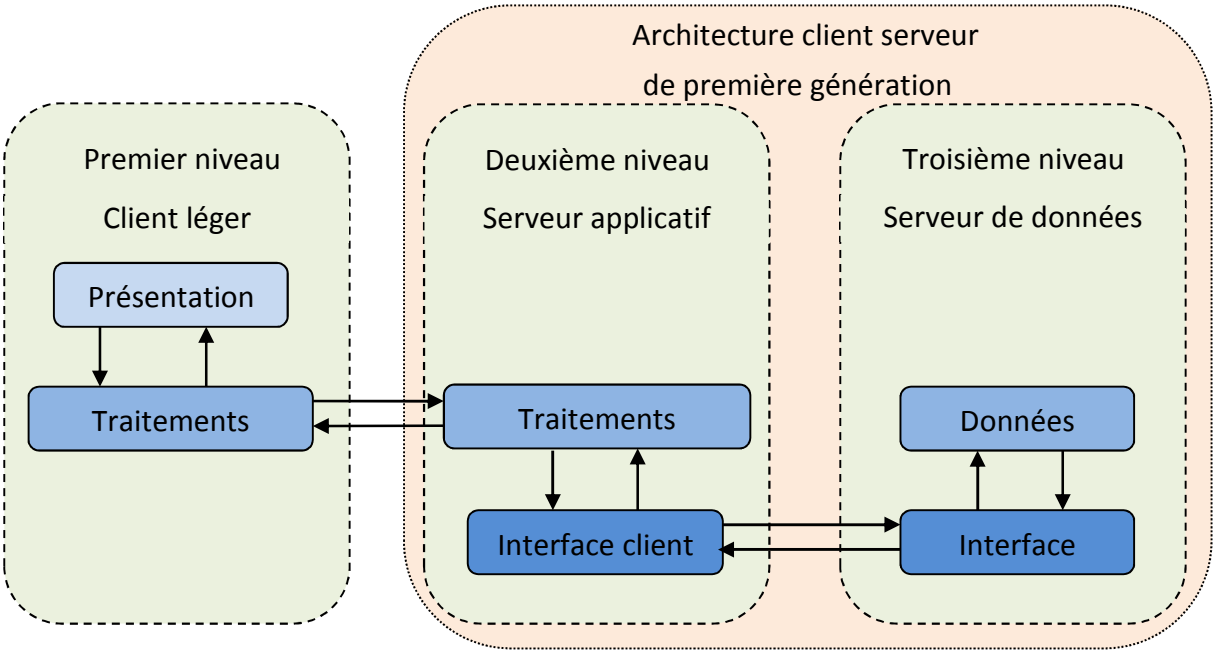


Figure 21 : Schéma d'une architecture 3-tiers

L'architecture trois tiers a corrigé les excès du client lourd en déportant une partie du traitement sur le serveur applicatif. Le poste client s'en trouve donc allégé. Cependant, le serveur applicatif se trouve plus fortement sollicité car il va concentrer l'ensemble des demandes des traitements globaux des clients. De par son rôle, il va devenir la pierre angulaire des applications, ce qui va soulever des problèmes de dimensionnement de serveur et de gestion de la montée en charge. Les contraintes semblent donc être inversées par rapport à l'architecture deux-tiers : le client est soulagé au détriment du serveur.

L'équilibre de charge semble être atteint avec la génération suivante : les architectures n-tiers.

4 Les applications n-tiers

Une architecture n-tiers continue à s'appuyer sur les trois niveaux définis pour les architectures trois-tiers. Cependant, afin de soulager la charge du serveur d'application, tel que défini dans l'architecture précédente, les services fournis ont été scindés en objets métiers spécialisés et indépendants. Ces objets ou composants rendent un service si possible générique, clairement identifié et ils sont capables de communiquer entre eux. Cette séparation permet donc de distribuer de multiples services sur des serveurs distincts [9], [10].

Pour gérer ces objets un environnement d'exploitation est nécessaire : le serveur d'objets [14]. Son rôle consiste à gérer la persistance des objets, les transactions d'objets métier, la montée en charge.

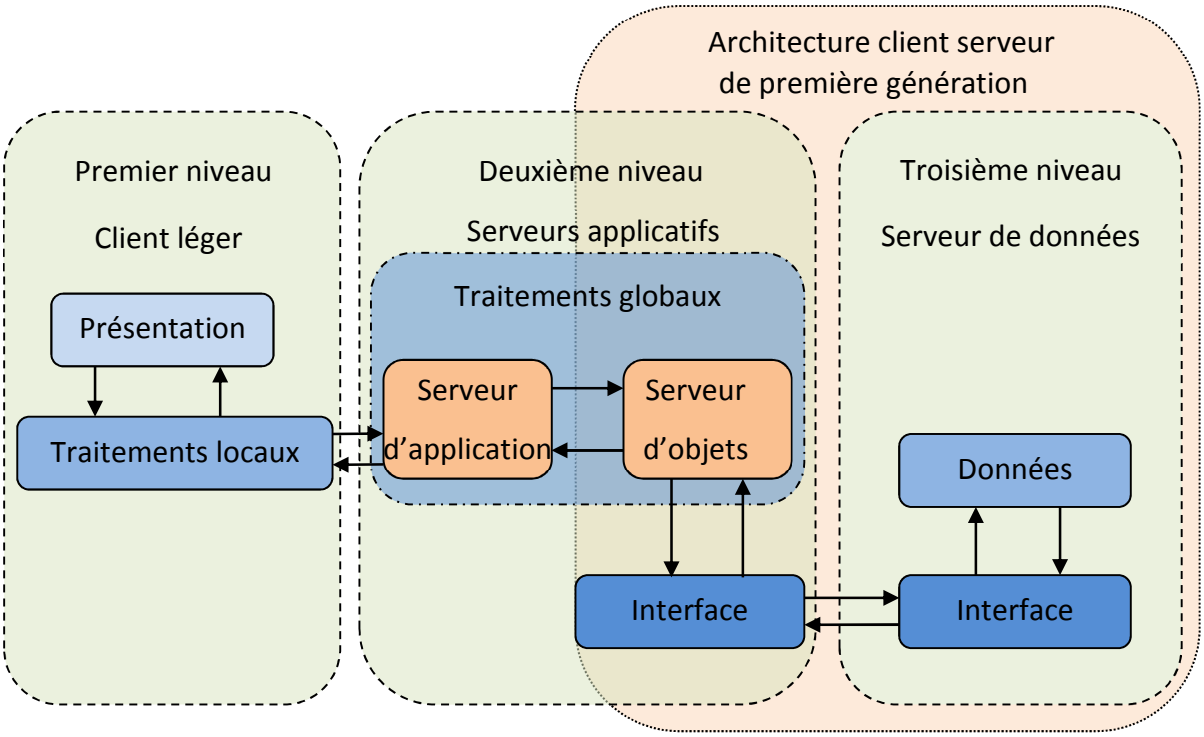


Figure 22 : Schéma d'une architecture n-tiers

Il s'agit à ce jour de l'architecture la plus flexible et la plus évoluée.

Bibliographie

[1] Microsoft, «.NET Framework Overview,» [En ligne]. Available: http://www.microsoft.com/net/overview.aspx. [Accès le 19 Avril 2011].

[2] Microsoft, «Platform Architecture,» [En ligne]. Available: http://technet.microsoft.com/fr-fr/library/aa683215(en-us).aspx. [Accès le 18 Octobre 2010].

[3] L. Piechocki, «UML, le langage de modélisation objet unifié,» 14 Septembre 2009. [En ligne]. Available: http://laurent-piechocki.developpez.com/uml/tutoriel/lp/cours/. [Accès le 20 Avril 2011].

[4] Rational Software, Microsoft, Hewlett-Packard, Oracle, Sterling Software, MCI Systemhouse, Unisys, ICON Computing, IntelliCorp, i-Logix, IBM, ObjecTime, Platinum Technology, Ptech, Taskon, Reich Technologies, Softeam, «UML v 1.1, Notation Guide,» 05 Aout 1997. [En ligne]. Available: http://www.omg.org/cgi-bin/apps/doclist.pl. [Accès le 01 Mars 2010].

[5] «BPMN,» [En ligne]. Available: http://en.q-bpm.org/mediawiki/index.php/BPMN. [Accès le 20 Mars 2010].

[6] JDN & JDN Solutions, «BPMN, la norme du BPM,» 3 Janvier 2008. [En ligne]. Available: http://www.journaldunet.com/solutions/acteurs/classement/08/bpms/0207-norme-bpmn.shtml. [Accès le 20 Mars 2010].

[7] OMG, «Business Process Model and Notation (BPMN),» [En ligne]. Available: http://www.omg.org/spec/BPMN/. [Accès le 20 Mars 2010].

[8] C. BLANLOEIL, «Evolution d'un système décisionnel propriétaire vers un système d'information décisionnel (SID),» TOURS, 2010.

[9] R. LEBLOND, «Vers une architecture n-tiers (Oral probatoire CNAM),» 02/12/1999.

[10] Hywan, «Annexe B. Introduction à l'architecture n-tiers,» [En ligne]. Available: http://hoa-project.net/Manuel/Reference/Annexe_Introduction_a_l-architecture_n-tiers.html. [Accès le 18 Octobre 2010].

[11] A. Bray et A. Redouly, «Le modèle OSI de l'ISO,» 23 Septembre 2010. [En ligne]. Available: http://www.opendoc.net/cours-modele-osi. [Accès le 18 Avril 2011].

[12] «Architecture trois tiers,» 25 Mars 2011. [En ligne]. Available: http://fr.wikipedia.org/wiki/Architecture_trois_tiers. [Accès le 25 Mars 2011].

[13] J. Niccolai, «virtualization-new-mainframe-vmware-says-497,» 21 04 2009. [En ligne]. Available: http://www.infoworld.com/d/virtualization/virtualization-new-mainframe-vmware-says-497?page=0,1. [Accès le 20 Octobre 2010].

[14] H. Makhmara, F. Niño et J.-L. Boichard, «Les Serveurs d'applications dans les Architectures N-Tiers,» 26 Juin 2002. [En ligne]. Available: http://medias.obs-mip.fr/www/Activite/Formation/Informatique/sgbd/cours013.html. [Accès le 18 Octobre 2010].

[15] D. Caromel, L. Mestre et R. Quillici, *Programmation Répartie et Architecture N-tiers,* Nice: Maîtrise informatique, Université Nice Sophia Antipolis, 2004.

Table des figures

Figure 1 : Organigramme général de la société ALFA Informatique 5
Figure 2 : Schéma de l'architecture de *Microsoft CRM* [2] 10
Figure 3 : Cartographie simplifiée des liens entre applications du système d'information 14
Figure 4 : Diagramme de PERT 20
Figure 5 : Architecture de la plateforme de test 29
Figure 6 : Nouvelle architecture de la plateforme de test 30
Figure 7 : Ensemble des écrans permettant d'accéder au script de chargement du formulaire de l'entité Compte dans *Microsoft CRM* 3.0 31
Figure 8 : Liaisons entre projets au sein de la solution des personnalisations du logiciel de CRM 33
Figure 9 : Nouvelle organisation des projets au sein de la solution des personnalisations du logiciel de CRM 35
Figure 10 : Schéma simplifié du logiciel de reprise de données 38
Figure 11 : Etape 1 - Importation de données 39
Figure 12 : Etape 2 - Sélection des colonnes de clé primaire 40
Figure 13 : Etape 3 - Recherche des doublons 41
Figure 14 : Etape 4 - Récapitulatif avant traitement 41
Figure 15 : Etape 5 - Résultat de l'importation 42
Figure 16 : Schéma des liens entre les concepts de base 75
Figure 17 : Schéma des liens entre les chapitres principaux de *Selligent* 76
Figure 18 : Les trois niveaux d'une application informatique [9] 78
Figure 19 : Schéma d'une architecture Mainframe 78
Figure 20 : Schéma d'une architecture client-serveur 79
Figure 21 : Schéma d'une architecture 3-tiers 79
Figure 22 : Schéma d'une architecture n-tiers 79

Vu le Professeur Principal,

p.o M. Schneider

M. MISSON

Vu, le Professeur responsable de la filière,

M. SCHNEIDER

Vu le Directeur du Cnam d'Auvergne

P. CONIL

A Aubière, le 14 mai 2012

Evolution d'une application de CRM au sein d'une SSII

CHAPELON Frédéric

Mémoire d'ingénieur C.N.A.M
C.N.A.M d'Auvergne 13/06/2012

Résumé

La CRM représente à la fois un concept et un environnement applicatif. Le concept consiste à recentrer son activité autour du client afin de créer un véritable lien avec ce dernier. La CRM applicative sert de support à ces concepts. Elle peut être classée en trois sous-systèmes : la CRM opérationnelle, analytique et collaborative.

Le sujet de ce mémoire est de présenter le travail effectué dans le cadre d'un changement de logiciel de CRM opérationnelle, de la solution *Selligent* vers la solution *Microsoft CRM*. Pour ce faire une analyse des processus métier a été initiée en utilisant la méthode BPMN. Le nouveau logiciel de CRM a été personnalisé au travers de développement .Net et de JavaScripts et un utilitaire de reprise de données a été développé. Ce travail représente les premiers fondements d'un projet global visant à consolider les informations et les outils de la société en vue d'améliorer le service rendu au client.

Mots clés : CRM, Microsoft, *Selligent*, BPMN, .Net.

Abstract

CRM is both a concept and an application environment. The idea is to refocus its business activity on customer to create a real customer relationship with him. The CRM application is used to support these concepts. It can be classified into three sub-systems: operational, analytical and collaborative CRM.

The subject of this paper is to present the work that was done for changing the operational CRM's software solution from *Selligent* to *Microsoft CRM*'s solution. To do this an analysis of the business process was initiated by using BPMN's method. JavaScript and .Net developments have been done to customize the new CRM software and a data recovery utility has been developed. This work represents the foundations of a global project whose purpose is to consolidate company's information and tools for improving client's service.

Keyword: CRM, Microsoft, *Selligent*, BPMN, .Net.