RESUME

La startup INSBI (Institut Business Intelligence) est une entreprise nouvellement créer qui dans son (Service qu’on propose) a une liste de service qui tourne autour de solutions de digitalisation d’entreprises et Business intelligence. Dans le souci de se faire un nom sur le marché INSBI a produit et lancé son premier produit qui est Hosteline. Hosteline est une plateforme d’hôtellerie en ligne qui a pour cible les propriétaires d’établissements hôteliers mettent leurs locaux à la disposition des clients qui à leur tour peuvent réserver ces diffèrent locaux. Hosteline est un portail web qui fédère les hôtels et donne un accès comparatif aux clients de cette plateforme. A travers cette plateforme les clients d’hôtels auront des avantages tant sur les prix que sur les facilités que pourront offrir ces établissements hôtelier grâce à son programme de fidélité. Ce système lorsqu’il sera mis en production génèreras une quantité importante de donnée par son système transactionnel. Sachant à quel point ces données sont utiles mais n’informent pas suffisamment a l’état brut, la direction de INSBI souhait mettre en place un système décisionnel pour pouvoir exploiter ces données et en tirer les choix et décisions stratégiques pour faire grandir le produit en offrant une meilleur qualité de service à ses consommateurs. Ces là que nous intervenons afin de mettre en place un système de repporting qui serviras les données issues des systèmes opérationnel dans des formats lisible et facilement interprétable. Notre travail consistait à répondre à un besoin précis consigné dans un cahier avec des contraintes de couts et de délais bien définies. Dans ce mémoire nous mettrons en relief, la méthode de travail, l’évolution des travaux à IPLANS et le suivi qu’était le nôtre pendant toute la durée de notre stage. Ce travail nous a permis non seulement de mettre en pratique des enseignements reçus pendant notre formation, d’en apprendre davantage sur les méthodes de fonctionnement des entreprises et aussi sur des notions nouvelles.

PARTIE I : ETAT DE L’ART

CHAP 1 Synthèse Bibliographique

I Business Intelligence

* 1. Introduction

Dans le monde les grandes entreprises dans leurs activités journalières produisent une quantité importante de données qui a long terme devient en quantité astronomiques. Le besoin d’exploiter ces données a des fin de pilotage des entreprises a fait naitre le Business Intelligence encore appelé Intelligence Economique ou encore Informatique décisionnel qui permet d’étudier l’environnement de l’entreprise aux moyens des données qu’elle possède. Les données étant en grande quantité, il faut le nettoyer, les structures avant de les stocker dans ce qu’on a appelé Data Warehouse (Entrepôt de données en français).

Le concept de Data Warehouse, tel que connu aujourd’hui, est apparu pour la première fois en 1980 ; l’idée consistait alors à réaliser une base de données destinée exclusivement au processus décisionnel. Les nouveaux besoins de l’entreprise, les quantités importantes de données produites par les systèmes opérationnels et l’apparition des technologies aptes à sa mise en œuvre ont contribué à l’apparition du concept « Data Warehouse » comme support aux systèmes décisionnels.

* 1. Les systèmes décisionnels

L’entrepôt de données est au centre du système décisionnel et sa raison d’être est la mise en place de ces systèmes décisionnels. Nous allons ici rappeler quelques définitions qui serviront ont mieux expliqué la suite.

Selon Jean-Louis Le Moigne, « Le système d’information est l’ensemble des méthodes et moyens de recueil de contrôle et de distribution des informations nécessaires à l’exercice de l’activité en tout point de l’organisation. Il a pour fonction de produire et de mémoriser les informations, de l’activité du système opérant (système opérationnel), puis de les mettre à disposition du système de décision (système de pilotage) » [Le Moigne 1977].

De cette définition on retient que le système opérationnel et le système décisionnel sont des parties du système d’information.

Selon Wikipédia, « L’informatique décisionnelle est l'informatique à l'usage des décideurs et des dirigeants d'entreprises » [<https://fr.wikipedia.org/wiki/Informatique_décisionnelle> consulté le 15/06/2018].

* 1. DECISIONNEL VS TRANSACTIONNEL

Le tableau 1.1 résume des différences entre le système opérationnel et le système décisionnel. Critère

**par les données**

Orienté applications - Orienté thèmes et sujets

Situation instantanée - Situation historique

Donnée détaillées et codées non redondantes - Informations agrégées cohérentes souvent avec redondance

Données changeantes constamment - Informations stables et synchronisées dans le temps

Pas de référentiel commun Un référentiel unique

**L’usage**

Assure l’activité au quotidien - Permet l’analyse et la prise de décision

Pour les opérationnels - Pour les décideurs

Mises à jour et requêtes simples = Lecture unique et requêtes complexes transparentes

Temps de réponse immédiats - Temps de réponse moins critiques

Faibles volumes à chaque transaction - Large volume manipulé

Conçu pour la mise à jour - Conçue pour l’extraction

Usage maîtrisé - Usage aléatoire

Figure XX : Tableau comparatif entre les systèmes transactionnels et les systèmes décisionnels.

Ces différences font ressortir la nécessité de mettre en place un système répondant aux besoins décisionnels. Ce système n’est rien d’autre que le « Data Warehouse ».

**2 Le Data Warehouse**

**2.1 Définition**

Bill Inmon définit le Data Warehouse, dans son livre considéré comme étant la référence dans le domaine ‘’Building the Data Warehouse’’ [Inmon, 2002] comme suit:

« Le Data Warehouse est une collection de données orientées sujet, intégrées, non volatiles et évolutives dans le temps, organisées pour le support d’un processus d’aide à la décision. »

Les paragraphes suivants illustrent les caractéristiques citées dans la définition d’Inmon.

**Orienté sujet** : le Data Warehouse est organisé autour des sujets majeurs de l’entreprise, contrairement à l’approche transactionnelle utilisée dans les systèmes opérationnels, qui sont conçus autour d’applications et de fonctions telles que : cartes bancaires, solvabilité client…, les Data Warehouse sont organisés autour de sujets majeurs de l’entreprise tels que : clientèle, ventes, produits…. Cette organisation affecte forcément la conception et l’implémentation des données contenues dans le Data Warehouse. Le contenu en données et en relations entre elles diffère aussi. Dans un système opérationnel, les données sont essentiellement destinées à satisfaire un processus fonctionnel et obéit à des règles de gestion, alors que celles d’un Data Warehouse sont destinées à un processus analytique.

**Intégrée** : le Data Warehouse va intégrer des données en provenance de différentes sources. Cela nécessite la gestion de toute incohérence.

**Evolutives dans le temps :** Dans un système décisionnel il est important de conserver les différentes valeurs d’une donnée, cela permet les comparaisons et le suivi de l’évolution des valeurs dans le temps, alors que dans un système opérationnel la valeur d’une donnée est simplement mise à jour. Dans un Data Warehouse chaque valeur est associée à un moment

« Every key structure in the data warehouse contains - implicitly or explicitly -an element of time » [Inmon, 2000].

**Non volatiles :** c’est ce qui est, en quelque sorte la conséquence de l’historisation décrite précédemment. Une donnée dans un environnement opérationnel peut être mise à jour ou supprimée, de telles opérations n’existent pas dans un environnement Data Warehouse. Organisées pour le support d’un processus d’aide à la décision : Les données du Data Warehouse sont organisées de manière à permettre l’exécution des processus d’aide à la décision (Reporting, Data Mining…).

**2.2 Historique**

L’origine du concept *« Data Warehouse » D.W* (entrepôt de données en français) remonte aux années 80, durant lesquelles un intérêt croissant au système décisionnel a vu le jour, dû essentiellement à l’émergence des SGBD relationnel et la simplicité du modèle relationnel et la puissance offerte par le langage SQL, au début, le Data Warehouse n’était rien d’autre qu’une copie des données du système opérationnel prise de façon périodique, dédiée à un environnement de support à la prise de décision. Ainsi, les données étaient extraites du système opérationnel, stockées dans une nouvelle base de données «concept d’infocentre », le motif principal étant de répondre aux requêtes des décideurs sans pour autant altérer les performances des systèmes opérationnels. Le Data Warehouse, tel qu’on le connaît actuellement, n’est plus vu comme une copie ou un cumul de copies prises de façon périodique- des données du système opérationnel. Il est devenu une nouvelle source d’information, alimenté avec des données recueillies et consolidées des différentes sources internes et externes.

**2.3 ARCHITECTURE D’UN DATA WAREHOUSE**

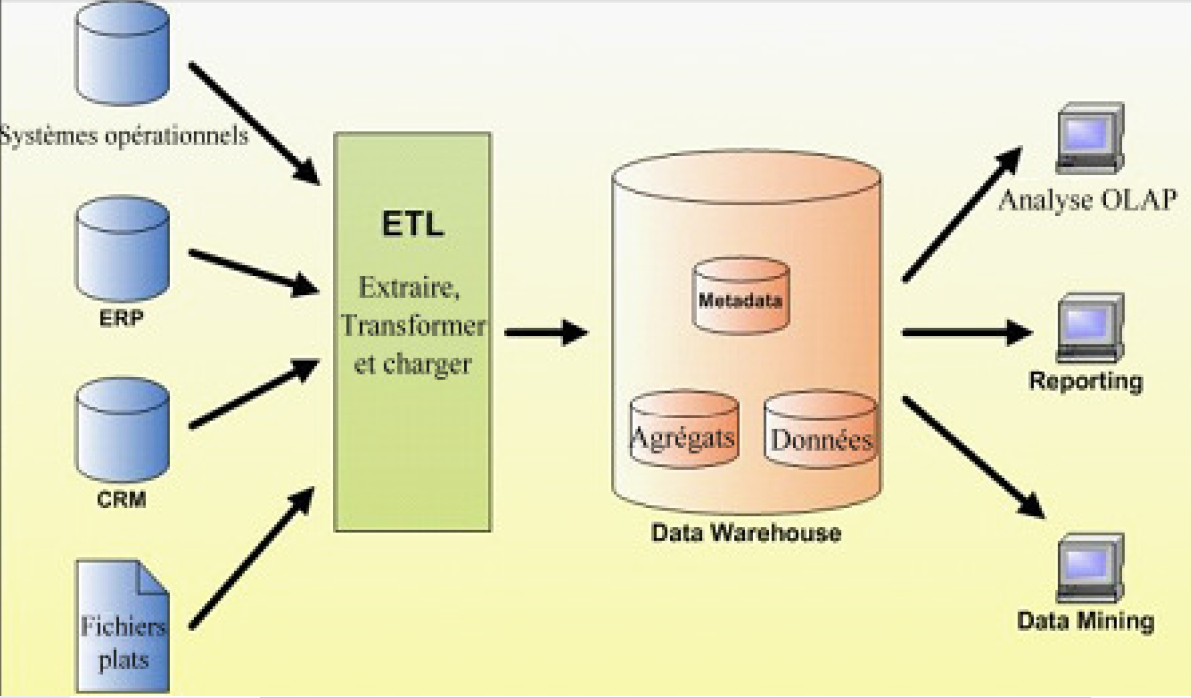


Figure XX Architecture generale d’un datawarehouse

La figure ci-dessus Illustre la forme générale d’un data Warehouse que nous allons détailler dans les paragraphes suivants.

**Les Sources de données** : Dans la figure, les représentations de systèmes opérationnel, ERP, CRM et Fichiers plat font office de source de données et c’est d’elles qu’on puisse les données pour alimenter la machine décisionnelle.

**ETL (Extract, Transform, Load):** C’est un ensemble de méthodes et d’outils qui servent a :

* Extract : Extraire les données de sources hétérogènes
* Transform : Transformation des données pour les mettre dans un format acceptable
* Load : Charger les données dans le data warehouse

**Data Warehouse :** L’unité de stockage des données. Il est constitué de plusieurs éléments dont :

* **Meta données :** ce sont les informations relatives à la structure des données, les méthodes d’agrégation et le lien entre les données opérationnelles et celles du Data Warehouse. Les métadonnées doivent renseigner sur :
  + Le modèle de données,
  + La structure des données telle qu’elle est vue par les développeurs,
  + La structure des données telle qu’elle est vue par les utilisateurs,
  + Les sources des données,
  + Les transformations nécessaires,
  + Suivi des alimentations,
* **Les Agrégats (Données agrégées) :** données agrégées à partir des données détaillées.

Les derniers éléments de la figure font partie de la phase d’exploitation du data warehouse et seront détaillés plus bas

1. **Modélisation des données de l’entrepôt**

**3.1 La modélisation dimensionnelle et ses concepts**

Les Data Warehouse sont destinés à la mise en place de systèmes décisionnels. Ces systèmes, devant répondre à des objectifs différents des systèmes transactionnels, ont fait ressortir très vite la nécessité de recourir à un modèle de données simplifié et aisément compréhensible. La modélisation dimensionnelle permet cela. Elle consiste à considérer un sujet d’analyse comme un cube à plusieurs dimensions, offrant des vues en tranches ou des analyses selon différents axes.

Figure cube.

**3.1.1 Le concept des Faits :** Une table de faits est la table centrale d’un modèle dimensionnel, où les mesures de performances sont stockées. Une ligne d’une table de faits correspond à une mesure. Ces mesures sont généralement des valeurs numériques, additives ; cependant des mesures textuelles peuvent exister mais sont rares. Le concepteur doit faire son possible pour faire des mesures textuelles des dimensions, car elles peuvent êtres corrélées efficacement avec les autres attributs textuels de dimensions.

**3.1.2 Le concept des Dimensions :** Les tables de dimension sont les tables qui raccompagnent une table de faits, elles contiennent les descriptions textuelles de l’activité. Une table de dimension est constituée de nombreuses colonnes qui décrivent une ligne. C’est grâce à cette table que l’entrepôt de données est compréhensible et utilisable; elles permettent des analyses en tranches et en dés. Une dimension est généralement constituée : d’une clé artificielle, une clé naturelle et des attributs.

**3.2 Différents modèles de la modélisation dimensionnelle**

**Modèle en étoile** : comme indiqué précédemment, ce modèle se présente comme une étoile dont le centre n’est autre que la table des faits et les branches sont les tables de dimension. La force de ce type de modélisation est sa lisibilité et sa performance.

**Modèle en flocon** : identique au modèle en étoile, sauf que ses branches sont éclatées en hiérarchies. Cette modélisation est généralement justifiée par l’économie d’espace de stockage, cependant elle peut s’avérer moins compréhensible pour l’utilisateur final, et très couteux en termes de performances.

**Modèle en constellation :** Ce n’est rien d’autre que plusieurs modèles en étoile liés entre eux par des dimensions communes.

3.3 La navigation dans les données

Une fois que le serveur OLAP a construit le cube multidimensionnel « ou simulé ce cube selon l’architecture du serveur », plusieurs opérations sont possibles sur ce dernier offrant ainsi la possibilité de naviguer dans les données qui le constituent. Ces opérations de navigation « Data Surfing » doivent être, d’une part, assez complexes pour adresser l’ensemble des données et, d’autre part, assez simples afin de permettre à l’utilisateur de circuler de manière libre et intuitive dans le modèle dimensionnel.

Afin de répondre à ces attentes, un ensemble de mécanismes est exploité, permettant une navigation par rapport à la dimension et par rapport à la granularité d’une dimension.

3.3.1 Slice & Dice

Le « Slicing » et le « Dicing » sont des techniques qui offrent la possibilité de faire des tranches « trancher » dans les données par rapport à des filtres de dimension bien précis, se classant de fait comme des opérations liées à la structure « se font sur les dimensions ». La différence entre eux se manifestent dans le fait que :

*Slicing is the process of retrieving a block of data from a cube by filtering on one dimension [Vincent Rainardi 2008].*

**Figure du Slicing**

*Dicing is the process of retrieving a block of data from a cube by filtering on all dimensions [Vincent Rainardi 2008].*

**Figure du dicing**

**4 Démarche de Construction d’un Data Warehouse**

Plusieurs chercheurs ou équipes de recherche ont essayé de proposer des démarches pour la réalisation d’un projet Data Warehouse, ces démarches se croisent essentiellement dans les étapes suivantes :

* Modélisation et conception du Data Warehouse,
* Alimentation du Data Warehouse,
* Mise en œuvre du Data Warehouse,
* Administration et maintenance du Data Warehouse,

**4.1 Modélisation et conception du Data Warehouse**

Les deux approches les plus connues dans la conception des Data Warehouse sont :

* L’approche basée sur les besoins d’analyse,
* L’approche basée sur les sources de données,

Aucune des deux approches citées n’est ni parfaite, ni applicable à tous les cas. Toutes deux doivent être étudiées pour choisir celle qui s’adapte le mieux à notre cas. Quel que soit l’approche adoptée pour la conception d’un Data Warehouse, la définition de celui-là reste la même. En étant un support d’aide à la décision, le Data Warehouse se base sur une architecture dimensionnelle.

**4.1.1 Approche « Besoins d’analyse »**

Le contenu du Data Warehouse sera déterminé selon les besoins de l’utilisateur final. Cette approche est aussi appelée « approche descendante » (Top-Down Approach) et est illustrée par R. Kimball grâce à son cycle de vie dimensionnel comme suit :

Figure

**Avantages**

Aucun risque de concevoir une solution obsolète avant d’être opérationnelle

Ndems

Pas de prise en compte de l’évolution des besoins de l’utilisateur.

Nécessite une modification de la structure du Data Warehouse en cas de nouveau besoin

Négligence du système opérationnel

Difficulté de déterminer les besoins des utilisateurs

**4.1.2 Approche « Source de données »**

Le contenu du Data Warehouse est déterminé selon les sources de données. Cette approche est appelée: Approche ascendante (Bottom-up Approach).

Figure

Avantages

Meilleure prise en charge de l’évolution des besoins

Ndems

Risque de concevoir une solution obsolète

avant qu’elle soit opérationnelle

Evolution du schéma des données source

Complexité de source de données

Inmon considère que l’utilisateur ne peut jamais déterminer ses besoins dès le départ, « *Donnez-moi ce que je vous demande, et je vous direz ce dont j’ai vraiment besoin* », il considère que les besoins sont en constante évolution.

**IV.1.3 Approche mixte**

Une combinaison des deux approches appelée hybride ou mixte peut s’avérer efficace. Elle prend en considération les sources de données et les besoins des utilisateurs.

Cette approche consiste à construire des schémas dimensionnels à partir des structures des données du système opérationnel, et les valider par rapport aux besoins analytiques. Cette approche cumule les avantages et quelques inconvénients des deux approches déjà citées, telles que la complexité des sources de données et la difficulté quant à la détermination des besoins analytiques.

**IV.2.1 Les phases de l’alimentation « E.T.L. »**

Les phases du processus E.T.L. représentent la mécanique d’alimentation du Data

Warehouse. Ainsi elles se déroulent comme suit :

**a) L’extraction des données**

*« L’extraction est la première étape du processus d’apport de données à l’entrepôt de données. Extraire, cela veut dire lire et interpréter les données sources et les copier dans la zone de préparation en vue de manipulations ultérieures.* » **[Kimball, 2005].**

Elle consiste en :

* Cibler les données,
* Appliquer les filtres nécessaires,
* Définir la fréquence de chargement

Lors du chargement des données, il faut extraire les nouvelles données ainsi que les changements intervenus sur ces données. Pour cela, il existe trois stratégies de capture de changement :

* **Colonnes d’audit :** la colonne d’audit, est une colonne qui enregistre la date d’insertion ou du dernier changement d’un enregistrement. Cette colonne est mise à jour soit par des triggers ou par les applications opérationnelles, d’où la nécessité de vérifier leur fiabilité.
* **Capture des logs :** certains outils ETL utilisent les fichiers logs des systèmes sources afin de détecter les changements (généralement logs du SGBD). En plus de l’absence de cette fonctionnalité sur certains outils ETL du marché, l’effacement des fichiers logs engendre la perte de toute information relative aux transactions.
* **Comparaison avec le dernier chargement :** le processus d’extraction sauvegarde des copies des chargements antérieurs, de manière à procéder à une comparaison lors de chaque nouvelle extraction. Il est impossible de rater un nouvel enregistrement avec cette méthode.

**b) La transformation des données**

La transformation est la seconde phase du processus. Cette étape, qui du reste est très importante, assure en réalité plusieurs tâches qui garantissent la fiabilité des données et leurs qualités. Ces tâches sont :

* Consolidation des données.
* Correction des données et élimination de toute ambiguïté.
* Elimination des données redondantes.
* Compléter et renseigner les valeurs manquantes. Cette opération se solde par la production d’informations dignes d’intérêt pour l’entreprise et de et sont donc prêtes à être entreposées.

**c) Le chargement des données**

C’est la dernière phase de l’alimentation d’un entrepôt de données, le chargement est une étape indispensable. Elle reste toutefois très délicate et exige une certaine connaissance des structures du système de gestion de la base de données (tables et index) afin d’optimiser au mieux le processus.

**IV.2.2 Politiques de l’alimentation**

Le processus de l’alimentation peut se faire de différentes manières. Le choix de la politique de chargement dépend des sources : disponibilité et accessibilité. Ces politiques

sont8 :

* **Push :** dans cette méthode, la logique de chargement est dans le système de production. Il " pousse " les données vers la zone de préparation quand il en a l'occasion. L'inconvénient est que si le système est occupé, il ne poussera jamais les données.
* **Pull :** contrairement de la méthode précédente, le Pull " tire " les données de la source vers la zone de préparation. L'inconvénient de cette méthode est qu'elle peut surcharger le système s'il est en cours d'utilisation.
* **Push-pull :** c'est la combinaison des deux méthodes. La source prépare les données à envoyer et indique à la zone de préparation qu'elle est prête. La zone de préparation va alors récupérer les données.

Les Processus ETL doit respecter les critères suivants

**Sûr :** assure l’acheminement des données et leur livraison.

**Rapide :** la quantité de données manipulées peut causer des lenteurs. Le processus d’alimentation doit palier à ce problème et assurer le chargement du Data Warehouse dans des délais acceptables.

**Correctif :** le processus d’alimentation doit apporter les correctifs nécessaires pour améliorer la qualité des données.

**Transparent :** le processus de l’ETL doit être transparent afin d’améliorer la qualité des données.

**IV.3 Mise en œuvre du Data Warehouse**

C’est la dernière étape d’un projet Data Warehouse, soit son exploitation. L’exploitation du Data Warehouse se fait par le biais d’un ensemble d’outils analytiques développés autour du Data Warehouse. Donc cette étape nécessite l’achèvement du développement, ou de la mise en place, de ces outils qui peuvent accomplir les fonctions suivantes:

**a. Requêtage ad-hoc :**

Le requêtage ad-hoc reste très fréquent dans ce type de projet. En effet, les utilisateurs de l’entrepôt de données, et spécialement les analystes, seront amenés à interagir avec le DW via des requêtes ad-hoc dans le but de faire les analyses requises par leurs métiers et, d’élaborer aussi, des rapports et des tableaux de bords spécifiques. L’accès à ce genre de service peut se faire via différentes méthodes et outils. Cependant, les spécialistes en la matière préconisent de laisser la possibilité à l’utilisateur de choisir les outils qui lui paraissent les plus adéquats.

**b. Reporting :**

Destiné essentiellement à la production de rapports et de tableaux de bord, *« il est la présentation périodique de rapports sur les activités et résultats d'une organisation, d'une unité de travail ou du responsable d'une fonction, destinée à en informer ceux chargés de les superviser en interne ou en externe, ou tout simplement concernés par ces activités ou résultants »9.*

Ces outils de Reporting ne sont pas, à proprement parler, des instruments d'aide à la décision, mais, lorsqu’ils sont utilisés de manière appropriée, ils peuvent fournir une précieuse vue d’ensemble.

Les rapports sont alors crées par le biais d’outils de Reporting qui permettent de leur donner un format prédéterminé. Les requêtes sont constituées lors de l’élaboration des rapports qui seront ensuite diffusés périodiquement en automatique ou ponctuellement à la demande.

**d. Tableaux de bord :**

Les tableaux de bord sont un outil de pilotage qui donne une vision sur l’évolution d’un processus, afin de permettre aux responsables de mettre en place des actions correctives.

« Le tableau de bord est un ensemble d’indicateurs peu nombreux conçus pour permettre aux gestionnaires de prendre connaissance de l’état et de l’évolution des systèmes qu’ils pilotent et d’identifier les tendances qui les influenceront sur un horizon cohérent avec la nature de leurs fonctions » **[Bouquin, 2003].**

Cette forme de restitution a la particularité de se limiter à l’essentiel, c'est-à-dire la mise en évidence de l’état d’un indicateur par rapport à un objectif, tout en adoptant une représentation graphique de l’information.

**e. Data Mining :**

Au sens littéral du terme, le Data Mining signifie le forage de données. Le but de ce forage est d’extraire de la matière brute qui, dans notre cas, représente de nouvelles connaissances. L’idée de départ veut qu’il existe dans toute entreprise des connaissances utiles, cachées sous des gisements de données. Le Data Mining permet donc, grâce à un certain nombre de techniques, de découvrir ces connaissances en faisant apparaître des corrélations entre ces données.

Le Data Warehouse constituera alors la première source de données sur laquelle s’exécutera le processus de découverte de connaissances. Dans la majeure partie du temps, l’entrepôt de données représente un pré requis indispensable à toute fouille de données.

Le recours à ce genre de méthode est de plus en plus utilisé dans les entreprises modernes. Les applications et outils implémentant ces solutions sont rarement développés en interne. En effet, les entreprises préfèrent se reposer sur des valeurs sûres du marché afin d’exploiter au plus vite les données en leur possession.

**IV.4 Maintenance et expansion**

La mise en service du Data Warehouse ne signifie pas la fin du projet, car un projet

Data Warehouse nécessite un suivi constant compte tenu des besoins d’optimisation de performance et ou d’expansion. Il est donc nécessaire d’investir dans les domaines suivants

**[Kimball, 2002]** :

**Support :** assurer un support aux utilisateurs pour leur faire apprécier l’utilisation de l’entrepôt de données. En outre, la relation directe avec les utilisateurs permet de détecter les correctifs nécessaires à apporter.

**Formation :** il est indispensable d’offrir un programme de formation permanant aux utilisateurs de l’entrepôt de données.

**Support technique :** un entrepôt de données est considéré comme un environnement de production. Naturellement le support technique doit surveiller avec la plus grande vigilance les performances et les tendances en ce qui concerne la charge du système.

**Management de l’évolution :** il faut toujours s’assurer que l’implémentation répond aux besoins de l’entreprise. Les revues systématiques à certain point de contrôle sont un outil clé pour détecter et définir les possibilités d’amélioration. En plus du suivi et de la maintenance du Data Warehouse, des demandes d’expansion sont envisageables pour de nouveaux besoins, de nouvelles données ou pour des améliorations.

Ces travaux d’expansion sont à prévoir de façon à faciliter l’évolution du schéma du

Data Warehouse.

CHAP 2 Présentation de Hosteline

* 1. Présentation Technique de Hosteline

Comme la majorité des projets informatiques, Hosteline a été conçu et réalisé selon un canevas avec des langages et méthodes de conception bien connu. En effet ce projet a été piloté par la méthodologie agile SCRUM (Annexe xx) avec le langage UML pour accompagne la conception. Apres une description brève du langage UML, Les diagrammes qui ont servi à la conception de Hosteline seront présentés et commentés

* + 1. Le langage UML

**UML** (Unified Modeling Language, que l’on peut traduire par langage de modélisation unifié) est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, et est devenu désormais la référence en terme de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet. La modélisation consiste à créer une représentation simplifiée d’un problème : le modèle. Grâce au modèle il est possible de représenter simplement un problème, un concept et le simuler. La modélisation comporte deux composantes :

* L’analyse, c’est-à-dire l’étude du problème
* La conception, soit la mise au point d’une solution au problème

Le méta modèle UML fournit une panoplie d’outils permettant de représenter l’ensemble des éléments du monde objet (classes, objets, ...) ainsi que les liens qui les relie. Toutefois, étant donné qu’une seule représentation est trop subjective, UML fournit un moyen astucieux permettant de représenter diverses projections d’une même représentation grâce aux vues. Une vue est constituée d’un ou plusieurs diagrammes. On distingue deux types de vues :

* Les **vues statiques**, c’est-à-dire représentant le système physiquement
  + diagrammes d’objets
  + diagrammes de classes
  + diagrammes de cas d’utilisation
  + diagrammes de composants
  + diagrammes de déploiement
* Les **vues dynamiques**, montrant le fonctionnement du système
  + diagrammes de séquence
  + diagrammes de collaboration
  + diagrammes d’états-transitions
  + diagrammes d’activités

Mais en ce qui concerne notre cas d’étude présent, nous ne représenterons que les diagrammes utilisées lors de la conception de la plateforme Hosteline.

* + 1. Le diagramme de cas d’utilisation

Les **diagrammes de cas d'utilisation** sont utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisation (use cases) [https://fr.wikipedia.org/wiki/Diagramme\_des\_cas\_d%27utilisation consulté le 16/06/2018].

Le tableau suivant présente la liste des formalismes qui sont utilisé dans la représentation des diagrammes de classe.

Tableau

L’étude de Hosteline dans la partie diagramme de class à donner deux diagrammes représentent des groupes d’acteurs différents et ayant peu ou pas d’actions en commun

Figure: Use case 1

Figure: Use case 2

* + 1. Le diagramme de Classes

Le **diagramme de classes** est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

Le tableau suivant présente les différents formalismes utilisés dans les représentations des diagrammes de classes.

Tableau

En ce qui concerne Hosteline, dans sa phase d’étude le diagramme de classe qui en est sorti est celui de la figure suivante (Version agrandie en annexe XX).

Figure Class diagramme

* + 1. Le diagramme de déploiement

Le **diagramme de déploiement** est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les **composants** du système sont répartis ainsi que leurs relations entre eux. Les éléments utilisés par un **diagramme de déploiement** sont principalement les **nœuds**, les **composants**, les **associations** et les **artefacts**. Les caractéristiques des ressources matérielles physiques et des supports de communication peuvent être précisées par stéréotype

Tableau Formalismes.

La figure suivante est le diagramme de déploiement qui découle de l’analyse UML de Hosteline.

Figure diagramme de déploiement

* 1. LE SYSTEME OPERATIONNEL ACTUEL

Si on parle de mettre en œuvre un système décisionnel sur Hosteline c’est parce que un système opérationnel qui va fournir les données a ce dernier existe déjà et est tout à fait fonctionnel. Dans cette partie nous allons présenter en quelques figure des interfaces du Systèmes opérationnel par les quels la collecte de données est plus forte sur Hosteline.