

Rapport de stage de fin d'études

Intégration d'un module de Business Intelligence dans un ERP

Société Objectif-PI

Barnabas Csomor

Février 2017 - Juillet 2017

Université Grenoble Alpes

Master 2 Mathématiques et Informatique Appliquées aux Sciences Humaines et Sociales

Parcours Web, Informatique et Connaissance

Tutrice de stage : Julie DUGDALE

Maître de stage : Sylvain FEROT

Remerciements

Je tiens, dans un premier temps, à remercier M. Sylvain FEROT responsable R&D, mon maître de stage pour toute son aide, ses bons conseils et le temps qu'il a consacré pour ma formation.

Je remercie également M. Raphaël MAUDET, gérant de la société Objectif-PI, de m'avoir accueilli au sein de sa société pour mon stage de fin d'étude. Ils m'ont permis de faire de ce stage, une excellente expérience sur un plan professionnel et humain.

Je remercie également M. Arthur PARISI, responsable développement, M. Thomas GONZALEZ et M. Bastien MARQUE, développeurs fonctionnels pour leurs aides et le temps qu'ils m'ont consacré afin de pouvoir avancer dans mes projets et de mener à bonne fin l'ensemble des objectifs de mon stage.

Je remercie le corps enseignant du master ainsi que Julie DUGDALE ma tutrice de stage de nous avoir transmis leurs connaissances que j'ai pu mettre en pratique lors du stage. Merci à l'université Grenoble Alpes de nous offrir la possibilité de faire ce projet de fin d'étude, de nous donner ainsi ce privilège de travailler sur des projets très intéressants, très formateurs sur le monde de l'entreprise et ainsi de commencer à se forger un début de carrière professionnel.

Résumé

Ce rapport de stage de fin d'étude traite sur les différentes phases et les problématiques de la mise en place d'un module de Business Intelligence dans un ERP.

L'objectif est de proposer des tableaux de bord interactifs aux utilisateurs regroupant chacun un des secteurs de l'ERP : la vente, la production et les achats de l'entreprise. Le but est de fournir des indicateurs pertinents aux utilisateurs afin de les aider dans la prise de décision et de leur offrir une vue d'ensemble des données de l'ERP.

Durant ce stage j'ai effectué des travaux de recherche sur les outils BI présents sur le marché. Le choix de l'outil à utiliser lors du stage s'est porté sur la suite décisionnelle de Pentaho.

Ce rapport aborde les différentes phases du projet : la mise en place d'un entrepôt de donnée et du processus d'ETL pour l'alimenter ainsi que de la modélisation, la performance et la conception des cubes de données pour l'analyse multidimensionnel, la mise en place d'un serveur BI, la modélisation et la conception des tableaux de bord ainsi que leur intégration à l'ERP.

Mots clés : Pentaho, ERP Open Source, Business Intelligence, Tableau de bord

Abstract

This report of end-of-study internship deals with the different phases and the problems of setting up a module of Business Intelligence in an ERP.

The aim is to provide interactive dashboards to our users, each one of them grouping one of the sectors of the ERP : sales, productions and purchases of the company. The goal is to provide relevant indicators to our users to assist them in decision-making and to provide them an overview of data which is in the ERP.

During my internship I did some research work on the BI tools present on the market, our choice was to use Pentaho to develop our dashboards.

This report covers the different phases of the project: the implementation of a data warehouse and the ETL process for its daily update, the modeling, performance and design of data cubes for multidimensional analysis, the installation of a BI server, the modeling and design of the dashboards as well as their integration in the ERP.

Key words : Pentaho, Open Source ERP, Business Intelligence, Dashboard

Sommaire

Remerciements	1
Résumé	2
Abstract	2
Sommaire	3
1. Introduction	5
2. Context du stage - Présentation de la société Objectif-PI	6
2.1 Son histoire jusqu'à aujourd'hui	6
2.2 Son organisation et Ses activités	7
3. Open-prod une ERP open source	8
4. L'équipe de travail et gestion du projet	9
5. Cahier des Charges	10
5.1 Objectifs	10
5.2 Fonctionnalités	11
5.3 Les tableau de bord	11
6. Etude de l'existant	12
7. Technologie utilisé:	13
7.1 Travaux de recherche d'un outil de BI intégrable à l'ERP	13
7.2 Pentaho	16
7.2.1 ETL - Kettle	16
7.2.2 Pentaho Community Dashboard Editor	16
8. Conception et réalisation	19
8.1 La partie ETL	19
8.1.1 Extraction des données	19
8.1.2 Transformations	20
8.1.3 Mise-à-jour de l'entrepôt de données	21
8.1.4 Conception des cubes	21
8.2 Les tableaux de bord	24
8.2.1 Internationalisation des tableaux de bord	24
8.2.2 Conversion au format PDF des tableaux de bords	24
8.2.3 Mise en place d'une Connection Pool	25
8.2.4 Intégration de pentaho dans l'ERP	25
8.2.5 Authentification sur le serveur BI	25
9. Tests	27

10. Conclusion	28
11. Bibliographie	29
12. Webographie	29
13. Liste des illustrations	30
14. Acronymes	30
Annexes	31
Annexe I - Description des cinq niveaux de Business Intelligence intégrable dans un ERP.	32
Annexe II - Schéma et indicateurs des tableaux de bord	41
Annexe II - Rendu PDF des tableaux de bord	59

1. Introduction

Dans le cadre de ma formation en master Mathématiques et Informatique Appliquées aux Sciences Humaines et Sociales dans le parcours Web, Informatique et Connaissance à l'Université Grenoble Alpes, j'ai effectué mon stage de fin d'étude de six mois au sein de la société Objectif-PI à Eybens (38), dans la région grenobloise. Ma mission fut de développer un module de Business Intelligence pour l'ERP de la société.

Les entreprises qui utilisent les ERPs ont de grandes quantités de données qui cachent des informations essentielles pour l'aide à la prise de décision et dont il est difficile d'avoir une vue d'ensemble. Ajouter de la Business Intelligence à l'ERP comporte de nombreux avantages, cela permet de présenter de manière structurée et cohérente les informations, d'analyser les données de l'entreprise, de faciliter la prise de décision grâce à des indicateurs pertinents et d'anticiper et prévoir les tendances. On peut parler de cinq niveaux de Business Intelligence dans une application ERP, comme vous pouvez le lire en annexe I. Ce rapport traite sur les tableaux de bords avec beaucoup d'interaction de la part de l'utilisateur pour une aide à la prise décision.

Ce rapport porte sur le développement de tableaux de bords et de leur intégration dans l'ERP ainsi que du processus de l'ETL pour alimenter l'entrepôt de données servant pour la BI. J'aborde notamment les différents problématiques pour la mise en place d'une solution de Business Intelligence au sein de l'ERP comme le choix de la technologie à utiliser pour une grande quantité de données, l'intégration transparente pour l'utilisateur d'une solution décisionnel au sein de l'ERP.

2. Context du stage - Présentation de la société Objectif-PI

Objectif-PI est une société d'intégration et d'édition d'ERP dans le domaine de l'industrie. Avec plus de 8 ans d'expérience, elle s'est spécialisée dans les domaines des Systèmes d'information, des ERPs et de la production. Composée d'une dizaine d'employé, Objectif-PI a réalisé en 2016 un chiffre d'affaires de 800 000 €.

2.1. Son histoire jusqu'à aujourd'hui

Création de la société

Objectif-PI a été créé en 2009 par M. Raphaël MAUDET dans la région grenobloise (à Lancey – 38 190). Après plus de 15 ans d'activité dans le monde informatique, au sein de grands groupes français, son objectif était de débuter une activité dans les systèmes d'informations pour apporter son expérience et son savoir-faire auprès d'autres entreprises.

Premier projet

Objectif-PI débute son activité par un important projet d'un an et demi, au sein de STMicroelectronics. L'objectif était d'intégrer et d'implémenter un nouveau système d'information sur le site de production de Crolles (38 – Isère).

Les débuts dans les ERPs open-source

A l'époque, les ERPs open-source commençaient à voir le jour et à devenir une alternative intéressante pour les petites et moyennes entreprises. En effet, les leaders du marché des ERPs, les ERPs propriétaires, tels que *SAP*, *Oracle* ou encore *SAGE-Adonix* étaient et sont toujours des solutions onéreuses voir inadaptées pour de telles structures.

Odoo SA, entre autre, fit partie de ces précurseurs sur le marché des ERPs open-source, avec notamment leur solution : Open-ERP. Ainsi, en 2010, à la suite du projet STMicroelectronics, Objectif-PI se tourna vers ce secteur des ERPs open-source et plus particulièrement vers le produit Open-ERP. Concernant cette solution, elle débute par des activités de conseils et d'accompagnements : suivi de projet d'intégration, formations fonctionnelles et techniques...

Développement d'un ERP open-source

En 2011, face aux demandes et aux attentes clients, Objectif-PI décida de se lancer dans le développement d'ERP open-source et plus particulièrement d'un ERP pour le secteur industriel. La stratégie de la société fut de partir de la solution existante, Open-ERP, pour la restructurer, améliorer ses fonctionnalités existantes et lui apporter de nouvelles

fonctionnalités. L'objectif était de mettre en place un ERP open-source qui puisse répondre aux attentes de clients industriels par le développement, entre autre, d'une GPAO (Gestion de Production Assisté par Ordinateur) intégrée.

Pour en arriver à ces résultats, Objectif-PI mit l'accent sur une activité de développement informatique et moins sur celle de conseils et d'accompagnements. Durant cette période, elle développa son savoir-faire autour de la conception produit et du développement d'ERP. En effet, ces années ont été l'occasion de construire une équipe de développeur technique et de se donner les moyens de répondre à ces objectifs : concevoir et développer un ERP open source fiable et performant.

Bien sûr, tout développement de produit doit se faire au plus proche du terrain pour discerner au mieux à la fois les tendances des marchés mais surtout les besoins clients. C'est pourquoi, quelques projets d'intégration furent réalisés, durant cette période, à la fois pour la solution Open-ERP et pour leur nouvel ERP open source, Open-Prod. Ces projets permirent à la société d'avoir des fonds pour investir dans ses développements et, par ailleurs, de faire connaître son produit.

2.2. Son organisation et Ses activités

Objectif-PI est constitué de trois grands types de ressources : des concepteurs produit, des développeurs et des consultants/chefs de projet. Chacun a un rôle dans une ou plusieurs de ces activités. Ces dernières sont diverses mais reste dans le secteur de l'informatique et des systèmes d'informations.

Aujourd'hui, sur l'ensemble de ces activités, Objectif-PI en possède deux principales et majeures :

Conception et développement de système d'information

Très importante ces dernières années, cette activité requiert différentes ressources :

Des concepteurs pour concevoir un ERP cohérent dans son fonctionnement tout en apportant l'ensemble des fonctionnalités souhaitées par les clients.

Des développeurs pour concrétiser ce qui a été conçu et obtenir l'outil désiré

Cette activité a pour vocation de diminuer dans le futur

Intégration des ERPs de la société auprès de client

Cette activité moins importante ces dernières années, a vocation à augmenter au sein d'Objectif-PI. Elle requiert des **consultants/chefs de projet**. Ces ressources accompagnent les clients dans l'intégration de l'ERP au sein de leur structure. En effet, ce type de système reste complexe et requiert donc une certaine organisation pour que l'intégration se passe dans les meilleures conditions. Objectif-PI utilise des méthodes agiles telles que la méthode SCRUM pour gérer de tels projets.

3. Open-prod une ERP open source

Durant mon stage, j'ai travaillé sur Open-prod, un ERP open source écrit en python JavaScript et XML.

Un ERP (Enterprise Resource Planning) ou également appelé PGI (Progiciel de Gestion Intégré) est un système d'information qui permet de gérer et suivre au quotidien, l'ensemble des informations et des services opérationnels d'une entreprise.

Que signifie un ERP open-source ?

Pour Objectif-PI, open-source ne veut pas dire libre accès. Ses ERPs ne sont pas en téléchargement libre sur internet et accessible à tout le monde. En revanche, pour elle, open-source signifie que le produit n'a pas de coût de licence, d'une part, contrairement aux ERPs propriétaires, et surtout que son architecture et son code sont accessibles par toutes personnes en possession du produit. Ainsi tout client, ayant l'ERP, est à même de pouvoir modifier le code et la structure du logiciel. Objectif-PI choisit les clients auxquels elle propose ses produits, les forme et les suit une fois ceux-ci intégrés. L'objectif est de faciliter l'appropriation du produit par les clients pour qu'ils ne dénaturent pas son image, chose qui pourrait facilement arriver de par sa complexité.

Dans l'open-source, on retrouve également la notion de communauté. En effet, le but de l'open-source est de regrouper un ensemble d'experts sur un produit afin qu'ils apportent leurs contributions pour le faire évoluer. Objectif-PI travaille actuellement sur cette notion, notamment à travers la mise en place de partenaires autour de ses ERPs. Cet objectif a pour but d'améliorer la diffusion de ses solutions mais également d'augmenter leur nombre d'expert afin de les rendre plus pérennes.

Quels sont les avantages et inconvénients d'un ERP open source?

Un logiciel open-source est régit par 4 types de liberté : liberté d'utilisation, liberté d'étude, liberté d'amélioration et liberté de redistribution. Ouvrir le code d'un logiciel permet ainsi d'avoir une grande transparence quant au produit mais également de permettre sa personnalisation et son amélioration. Si nous nous rapportons aux ERPs, l'open-source apporte, entre autre, les avantages suivant :

L'ERP s'adapte à la structure dans lequel il est intégré et non l'inverse

Il y a une **indépendance vis-à-vis d'un éditeur** et de son avenir (faillite, fusion...)

Une qualité technique supérieure du produit, car la communauté construite autour du produit regroupe un ensemble d'experts du produit.

Un coût financier moindre par rapport à des ERPs propriétaire : pas de coûts de licences, pas de coûts additionnels pour des fonctionnalités supplémentaires....

Bien sûr, ce monde de l'open-source peut être moins positif, notamment si la communauté sur lequel se repose le produit n'est pas « solide » en termes d'évolutions, de technologies ou de maintenabilité du produit.

Enfin, pour Objectif-PI, le côté open source apporte un « leitmotiv » car il demande d'être toujours bon au niveau de ses produits et de ce monde du système d'information. En effet, si votre solution n'évolue pas, ne s'améliore pas, elle deviendra rapidement obsolète. De plus, le code étant accessible librement, cette volonté d'être toujours en évolution et bon sur son produit permet d'éviter de se faire « copier » ou de se faire prendre la solution. Bien sûr, il ne faut pas oublier que ces solutions sont très complexes et demande des connaissances pour pouvoir les vendre et les intégrer auprès de client.

Open-Prod est donc un ERP open source, flexible et modulaire entièrement dédié à la gestion de production.

4. L'équipe de travail et gestion du projet

J'ai travaillé en étroite collaboration avec M. Johan SALLEE, consultant fonctionnel, qui a établi la liste des indicateurs pour la BI et a réalisé les maquettes des tableaux de bords que j'ai eu à implémenter.

Mon maître de stage, M. Sylvain FEROT m'a formé sur le développement de modules pour l'ERP et m'a accompagné tout le long du projet en me conseillant sur beaucoup de points pratiques.

La société Objectif-PI suit la méthode agile pour son développement. Chaque matin l'équipe se réunissait pour faire un point sur ce qu'il avait fait la veille, ce qu'il projetait de faire le jour même et des difficultés rencontrés. À la fin de chaque cycle de développement, je faisais un bilan devant mon maître de stage et le directeur de la société. Cet échange permettait d'exposer le travail réalisé, de discuter sur la prochaine phase et si besoin, de modifier le plan de développement selon l'avancé du projet.

Le planning du projet :

- 01/02 - 06/02 : Formation à l'ERP d'Open-Prod
- 07/02 - 08/02: Définition des besoins et travaux de recherche sur les solutions possibles
- 09/02 - 10/02: Travaux de recherche et comparatif sur les différents outils BI disponible sur le marché
- 13/02 - 16/02: Découverte de pentaho et étude des possibilités d'implémentation dans l'ERP
- 17/02 - 24/02: Mise en place des cubes OLAP pour le tableau de bord des performances de vente - ETL
- 27/02 - 23/03: Tableau de bord performance de vente
- 24/03 - 28/03: Conversion PDF des tableaux de bords

- 29/03 - 31/03: Guide d'installation et de déploiement sur serveur
- 03/04 - 07/04: Tableau de bord force de vente
- 10/04 - 18/04: Intégration des tableaux de bords dans l'ERP, authentification sur le serveur Pentaho depuis l'ERP
- 19/04 - 21/04: Intégration du plugin Saiku
- 24/04 - 03/05: ETL pour le tableau de bord de production
- 04/05 - fin juin: Formation et transfert de connaissance sur l'ERP d'Open-prod
- juillet : tableaux de bord des achats et des productions.

On utilisait le logiciel de gestion de versions GitLab de la société Objectif-PI lors du projet.

5. Cahier des Charges

5.1. Objectifs

Les entreprises qui utilisent l'ERP ont de grandes quantités de données, de l'ordre de plusieurs millions de ventes et de productions. Du fait du volume important de ces données, il devient difficile d'avoir une vue d'ensemble, de leur donner un sens, de comprendre ce qu'elles expriment afin de prendre de bonnes décisions pour l'avenir de l'entreprise.

L'objectif est donc de proposer des tableaux de bords aux utilisateurs regroupant chacun un des secteurs de l'ERP : **la vente, la production et les achats** de l'entreprise.

Un tableau de bord est une forme particulière de rapport, particulièrement synthétique: tout doit tenir sur une feuille A4, ou sur un écran d'ordinateur pour faciliter la lecture des informations clés à l'utilisateur.

Les tableaux de bords présentent plusieurs indicateurs, qui ensemble offrent une représentation complète d'une activité de l'ERP.

On peut distinguer différents types d'indicateurs¹ :

- Les indicateurs d'alerte signalent la présence d'un dysfonctionnement, d'un état anormal impliquant une action corrective. Dans une gestion quotidienne, il peut s'agir des articles en rupture de stock, des besoins non transformés en commande...
- Les indicateurs d'efficience et d'équilibrage permettent de mesurer la situation actuelle par rapport aux objectifs fixés. Par exemple, l'état des articles ayant subi une baisse de consommation ou un faible taux de rotation (pour une révision éventuelle des données de planification).
- Les indicateurs d'anticipation affichent des renseignements sur les besoins futurs. Par exemple, un indicateur qui précisera l'influence qu'aura sur les stocks une activité prévue. Ces informations permettront au gestionnaire de stocks d'anticiper une situation en faisant varier les stocks à la hausse ou à la baisse.

¹ <http://www.silog.fr/solutions/silog-erp/silog-communication/bi/>

Les tableaux de bords doivent être personnalisables et interactifs, c'est-à-dire que chaque utilisateur doit avoir la possibilité d'appliquer des filtres, effectuer un drill-down selon les différents axes d'analyse, modifier le titre et cacher les indicateurs qu'il ne souhaite pas voir.

L'objectif est, grâce aux différents graphiques et aux indicateurs, d'offrir à l'utilisateur une vue d'ensemble des données, pour qu'il puisse à son tour en tirer les bonnes conclusions.

5.2. Fonctionnalités

Fonctionnalités communes à tous les tableaux de bord :

- Exporter les tableaux de bord au format PDF
- Exporter les images des graphiques au format image JPEG ou PNG
- Exporter les tableaux au format XLS
- Les tableaux de bord doivent avoir la possibilité d'utiliser des variables globales, par exemple pour la devise qui prendra la devise utilisée dans l'ERP
- L'utilisateur doit pouvoir modifier certains éléments du tableau de bord comme le titre et le sous-titre
- L'utilisateur doit pouvoir cacher les parties qui ne l'intéressent pas sur le tableau de bord
- Il faut que les tableaux de bords s'intègrent de manière transparente à l'utilisateur dans l'ERP
- L'utilisateur doit pouvoir filtrer, faire du drill-down, drill-up sur les cubes de données en fonction des axes d'analyses

5.3. Les tableau de bord

Les schémas des tableaux de bord avec leurs fonctionnalités et le détail des indicateurs sont disponibles dans l'annexe II, ce travail demandant des connaissances fonctionnelles poussé de l'ERP fut réalisé par M. Johan SALLEE consultant fonctionnel de la société.

6. Etude de l'existant

La BI faisant son apparition dans le monde des ERP, Odoo a également fait des développements sur le sujet et intégré un module de BI dans l'ERP. Comme l'explique la société Smile à la page 49 de son livre blanc sur les ERP, Odoo dispose d'un plugin nommé OpenObject qui permet d'introspecter les objets métier de l'ERP pour les exposer dans un cube OLAP et ainsi changer les axes et opérer des coupes dans le cube de données pour affiner les analyses. Ce qui permet à l'utilisateur de construire des tableaux de bord basiques, des graphiques et des tableaux croisés, comme on peut le voir sur les copies d'écran suivant :



	Total					Prix total devise société
	2016				+ 2017	
	+ Septembre	+ Octobre	+ Novembre	+ Décembre		
	Prix total devise société					
- Total	2 640,00	2 400,00	26 841,65	77 655,96	569 022,78	678 560,39
+ [P16] Desjoyaux Canada				888,11		888,11
+ [P19] Entrepôt Trévi			1 985,20			1 985,20
+ [P29] MAGASINS TREVI INC.					202 460,00	202 460,00
+ [P30] Magasins Trévi Inc. Laval			134,00			134,00
+ [P41] Piscine Okeanos Québec			127,65		67 927,00	68 054,65
+ [P45] SCP BOUCHERVILLE-275			10 310,40	27 312,00	987,54	38 609,94
+ [P47] SCP TERREBONNE-109			10 386,20	27 312,00	756,00	38 454,20
+ [P6] Bleu Electrique	2 640,00	2 400,00			31 825,00	36 865,00
+ [PA000019] SCP QUÉBEC-278			3 640,00	13 656,00		17 296,00
+ [PA000023] SCP WOODBRIDGE - 105			88,20		344,24	432,44
+ [PA000024] RADIANT POOLS			170,00			170,00

Ce module a l'avantage de tenir compte de la politique des droits de l'ERP, ainsi l'utilisateur qui n'a pas accès au module des ventes de l'ERP n'aura pas non plus accès aux données de vente dans le module BI. Les données étant celles de l'ERP, ils sont actualisés automatiquement, il n'y a donc aucune configuration à faire pour mettre en place un entrepôt de données. Les informations sont présentées de manière structurée et cohérente à l'utilisateur mais dès qu'on veut afficher les données de plusieurs objets qui ne sont pas liés

dans le même tableau de bord, effectuer des transformations, des vérifications sur les données et sortir du moule habituel, il faut créer un nouveau module Odoo qui vient étendre le module BI existant, ce qui est long, fastidieux et source d'erreur.

7. Technologie utilisé:

7.1. Travaux de recherche d'un outil de BI intégrable à l'ERP

Pour créer les tableaux de bord décrits dans la période précédente, le plugin OpenObject de l'ERP n'est pas assez complet. Venir compléter ce plugin serait réinventer la roue et demanderait un temps bien trop long de développement, c'est pourquoi j'ai effectué des travaux de recherche sur un outil BI intégrable à l'ERP.

Il y a un vaste choix d'outils BI sur le marché, wikipedia² en recense 80 et le site predictive analytics³ en recense 283.

J'ai basé mon comparatif des outils de BI sur plusieurs critères choix :

- Les possibilités d'intégration dans l'ERP : il faut que l'outil de BI puisse s'intégrer de façon transparente à l'utilisateur final dans un module de l'ERP.
- Le support : pour le développeur, plus la documentation, les tutoriels sont nombreux et la communauté grande et active moins il y a une perte de temps dans la recherche de solution au problème rencontré
- Le prix : ici encore le choix est vaste allant des solutions gratuites aux solutions payantes.

Généralement, on peut séparer le coût d'un outil BI en deux catégories: le coût d'achat de la licence initial et le coût de la maintenance et du support. La plupart des fournisseurs d'outils BI facturent selon ces deux types, à l'exception des entreprises qui vendent des solutions SaaS et Open Source, ces derniers proposent souvent un prix d'abonnement. L'abonnement nécessite un investissement initial moins élevé pour l'achat de la licence. Le modèle de prix peut aussi s'appliquer par utilisateur ou par serveur, ou les deux, avec des licences par utilisateurs pouvant atteindre plus de 10 000 euro par an.

- Les différents outils qui composent le logiciel : généralement les éditeurs ne proposent pas tous les outils (datamining, tableaux de bord, l'analyse ad-hoc, ETL...), les outils diffèrent également d'un éditeur à un autre, par exemple pour les tableaux de bords on va rarement retrouver le même design pour les graphes.

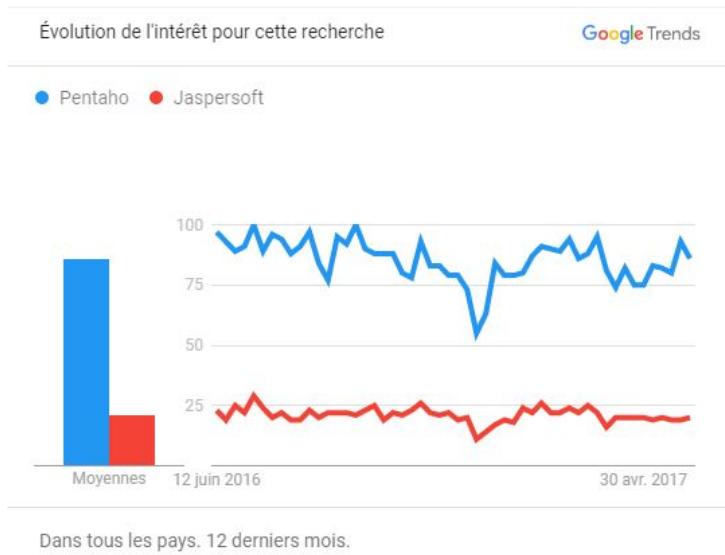
² https://en.wikipedia.org/wiki/Business_intelligence_software

³ <http://www.predictiveanalyticstoday.com/top-business-intelligence-tools/>

- Le code source du logiciel qui peut être Open Source ou propriétaire : l'ERP de la société Objectif-PI étant Open Source, la philosophie de la société vise à utiliser des outils Open Source si possible.
- La maintenabilité et l'évolution des outils à long terme.

En plus des nombreuses ressources de sites internet traitant sur le sujet, l'entreprise SMILE, un des géants de l'Open Source a publié deux livres blanc, un sur les outils BI Open Source et une autre sur les ERP Open Source. On trouve dans ce dernier, les progrès qui ont été fait dans les ERP Open Sources pour intégrer la business intelligence au sein des ERP.

Lors de la recherche, il était important de tenir compte de la date des ressources et des comparatifs, le secteur de la BI et des ERP a évolué rapidement durant les dernières années. Après de nombreuses recherches dans la littérature traitant sur le sujet, les deux solutions BI Open Source qui étaient supportés par la plus grande communauté et qui remplissaient les critères de choix étaient, la suite Pentaho et Jaspersoft. Ces deux éditeurs proposent une version communautaire et une version entreprise de leurs produit. La version communautaire est gratuite, tandis que la version d'entreprise est payante. Google trend indique plus de recherche sur le mot clé Pentaho que sur le mot clé Jaspersoft sur la dernière année :



La matrice comparative ci-dessous compare les 4 versions :

Critère	Pentaho Community Edition	Jaspersoft Community Edition	Pentaho Enterprise Edition	Jaspersoft Enterprise Edition
Intégration de façon transparente dans l'ERP	Oui	Oui	Oui	Oui
Analyse Ad-hoc (à la volée)	Plugin Saiku	Non	Oui	Oui
Export des résultats sous les principaux formats (PDF, Image, csv, xlsx)	Oui	Oui	Oui	Oui
Générateur de rapport Ad hoc	En phase de développement	Non	En phase de développement	Oui
Solution de tableau de bord	Oui, un large choix personnalisable	Non	Oui, un large choix personnalisable	Oui, un large choix personnalisable
Documentation en ligne	Bonne	Bonne	Très bonne	Très bonne
Communauté & forum	Communauté active	Communauté active	Communauté active + support	Communauté active + support
Prix	Gratuit	Gratuit	Payant sur mesure	Payant sur mesure
Licence	Licence Apache v2 ⁴	LGPL ⁵	Licence Commerciale	Licence Commerciale

Après consultation avec mon maître de stage et le directeur de la société, notre choix s'est porté sur la version communautaire de Pentaho qui intègre plus de possibilités que la version communautaire de Jasper et dont l'offre des possibilités de la version communautaire semble suffisant pour ce que l'on souhaite réaliser. Nous nous sommes fixés une période

⁴ https://fr.wikipedia.org/wiki/Licence_Apache

⁵ https://fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_limit%C3%A9e_GNU

d'essai de trois semaines pour la réalisation d'un tableau de bord basique et faire une tentative d'intégration à l'ERP.

7.2. Pentaho

Pentaho Community Edition 7.1 est la plate-forme décisionnelle utilisée pendant le stage. Pentaho couvre tous les secteurs de la BI avec ses différents logiciels, Pentaho reporting pour le reporting, Weka pour le data mining. Dans le cadre du stage, les logiciels utilisés sont :

- Kettle pour l'ETL
- Mondrian un serveur OLAP
- La suite CTools pour les tableaux de bords

7.2.1. ETL - Kettle

Afin d'alimenter notre entrepôt de données (le datawarehouse) à partir des différentes applications de l'entreprise, j'ai utilisé l'ETL Kettle de Pentaho, pour « Extract, Transform, Load ». Comme le nom l'indique, ces outils permettent d'extraire des données à partir de la base de données de l'ERP, de les transformer et de les charger dans la base de données cible, ici notre entrepôt de donnée servant pour la BI.

Un ETL permet d'éviter la réalisation de programmes batch répétitifs, souvent semblables, dont il faudra également assurer la maintenance. Pour une simplicité de déploiement et de maintenabilité sur les sites de production, il suffit de changer les données d'accès aux serveurs de l'ERP et de la BI dans un fichier de configuration.

7.2.2. Pentaho Community Dashboard Editor

Pentaho Community Dashboard Editor (CDE), fait partie de la suite CTools et permet la confection de tableaux de bords. Dans Pentaho CDE, la conception d'un tableau de bord est séparé en trois parties :

- Le layout : la structure du tableau de bord
- Les différents composants du dashboard : graphiques, tableaux, paramètres, filtres
- Les sources de données nécessaires à l'alimentation des composants, les data sources

Le Layout

Le Layout permet de positionner et d'agencer les différents composants de la page web. Il est important d'avoir une vue d'ensemble, un schéma du tableau de bord avant de commencer sa conception. Pentaho CDE repose sur le framework Bootstrap, avec une grille de 12 colonnes et une mise en page simplifiée. Il y a la possibilité de rajouter des fichiers ressources (JavaScript et CSS) ou des snippet (une petite portion réutilisable de code source ou de texte) permettant ainsi de personnaliser entièrement la mise en page.

Les composants

Les composants simplifient la construction d'un tableau de bord. Un composant est un simple objet Javascript qui encapsule toutes les propriétés et les comportements de l'objet permettant ainsi un degré de contrôle plus fin sur le composant du tableau de bord. Par exemple, un composant peut changer son comportement quand un changement a lieu dans le tableau de bord. Ces changements peuvent être définis avant, pendant et après l'exécution du composant. J'ai effectué des travaux de documentation sur le fonctionnement des composants, en traduisant la documentation disponible sur le site de Pentaho⁶ et en traduisant l'article sur les composants Javascripts écrit par Diethard Steiner⁷, dont voici le contenu : On peut ajouter, paramétriser, lier aux données et aux layouts les différents composants de notre tableau de bord grâce à l'interface graphique de CDE. Les composants sont les éléments centraux de notre tableau de bord.

Il y a 3 types de composants :

- Les composants **visuels** : ils sont affichés dans notre tableau de bord, comme par exemple du texte, des tableaux, des graphiques, des sélecteurs (des filtres, des calendriers)
- Les **paramètres** : ils représentent des valeurs qui sont partagés entre les composants, ils sont essentiels à l'interaction entre certains composants.
- Les **scripts** : des bouts de JavaScript permettant de personnaliser le rendu et le comportement des composants.

De manière générale, des composants vont réagir aux événements générés par l'utilisateur, comme par exemple un select, d'autres vont réagir aux changements des paramètres comme les composants graphiques et enfin d'autres vont réagir aux deux types de changements comme les tables. Un composant réagissant à un changement, modifie *son cycle de vie* dont l'exécution et le comportement sont contrôlés via des propriétés qui sont communs à la plupart des composants.

Pour résumer, un composant est initialisé au chargement du tableau de bord. Le composant check la valeur de la propriété *Execute At Start*, quand il a le feu vert, le composant entre dans la fonction *pre-execution* et exécute le code se trouvant à cet endroit. Par exemple dans cette phase, le composant peut émettre une requête spéciale ou aller chercher le contenu d'un paramètre pour l'utiliser ensuite. Le composant va ensuite se mettre à jour en entrant dans la fonction *post-execution*.

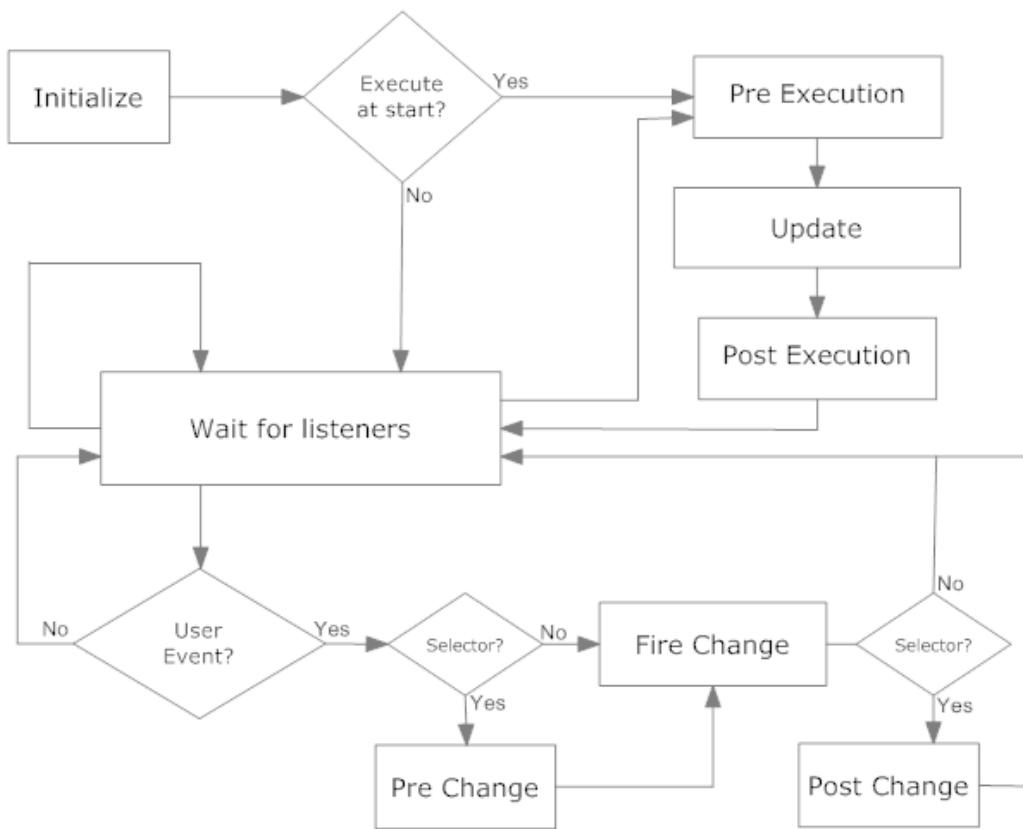
Ayant fini ce cycle, le composant attend pour des changements dans les paramètres qu'il écoute. Un changement à un de ces paramètres va déclencher les phases de *pre-execute*, *update*, et *post-execute*.

Quand un paramètre est changé, le composant détermine s'il s'agit d'un changement de la part de l'utilisateur, si c'est le cas et que le changement ne provient pas d'un sélecteur, un *fire change* se produit. L'information que le paramètre a changé va se propager dans le tableau de bord et chaque composant écoutant ce paramètre va se mettre à jour.

⁶ https://help.pentaho.com/Documentation/7.1/0R0/CTools/CDE_Dashboard_Overview

⁷ <http://diethardsteiner.github.io/ccc/2015/03/29/CCC-Core-Concepts.html>

Si le changement provient d'un sélecteur, le composant va exécuter toutes les instructions de *pre-change* puis de *post-change* après déclenchement du changement. Ce cycle de vie est illustré ci-dessous :



En venant modifier les bonnes phases du cycle de vie de l'objet, on peut modifier les composants, faire du drill-down, drill-up selon différents axes d'analyses lors de cliques à la souris, injecter les traductions, effectuer des modifications sur les données.

Dans un tableau de bord, il arrive souvent que certains composants dépendent d'autres composants, ainsi il faut aussi tenir compte de l'ordre de l'exécution des composants. Pentaho permet d'affecter une priorité à chaque composant dans l'ordre d'exécution. Ainsi on peut initialiser les composants qui ne dépendent pas entre eux en parallèle et mettre en place un ordre d'exécution pour les autres.

Les data sources

Pentaho dispose de beaucoup de possibilités pour ajouter des sources de données à notre tableau de bord. J'ai réalisé une première version du tableau de bord des performances de ventes avec un cube mondrian et le langage MDX.

“MDX (Multi Dimensional eXpression) est un langage de requêtes pour les bases de données multidimensionnelles, de la même manière que SQL est utilisé pour les requêtes sur les bases de données relationnelles. Dans son approche, MDX est proche du SQL sur son aspect select et where même si la similarité ne va pas plus loin. Le but des expressions

multidimensionnelles MDX est de rendre ais   et intuitif l'acc  s aux donn  es de diff  rentes dimensions. ⁸

Ne connaissant pas le langage MDX, j'ai ´etudi   ce langage et ´ecrit de la documentation ainsi qu'un tutoriel pour la soci  t  .

Lors de tests avec beaucoup de donn  es, (de l'ordre de plusieurs milliers de produits de clients et de lignes de factures extraites de bases de donn  es faisant plusieurs GB), il s'est av  r   que la combinaison Mondrian & MDX mettait plusieurs minutes ´a fournir les donn  es pour le tableau de bord. M  me si les donn  es ´etaient gard  es en cache dans le cube Mondrian sur le serveur BI de Pentaho et que la m  me requ  te par la suite ´etait bien plus rapide, ce d  lai ´etait bien trop grand au redémarrage du serveur et lors de la mise ´a jour des donn  es quand il fallait vider le cache. Malgr   avoir suivi les instructions sur divers forums, les indications sur le site de Pentaho⁹ et cr  er des tables d'agr  gats la performance n'augmenta pas de beaucoup pour les tables de faits. Je suis donc reparti pour une combinaison PostgreSQL & Pentaho, en utilisant le syst  me de cache de Pentaho¹⁰. Cette solution s'av  ra bien plus efficace, car on garde le syst  me de cache mais au d  marrage du serveur, on ne vient pas compl  ter le cube avec des valeurs par d  faut. Pour mieux comprendre, prenons l'exemple d'un cube de vente o   notre mesure est le chiffre d'affaires d'un produit qui n'a ´et   vendu qu'une seule fois et qu'on analyse notre cube selon les axes des clients, des vendeurs, du temps, des commandes, des adresses, des comptes analytiques et qu'on vienne faire un filtre sur ce produit. Mondrian vient compl  ter par des 0 dans le cube pour tous les jours, tous les clients, vendeurs, commandes, comptes analytiques et adresses o   le produit n'a pas ´et   vendu. D  s qu'on a des millions de ventes, cela devient lent et on se retrouve vite avec des cubes Mondrian de plusieurs GB en cache.

8. Conception et r  alisation

8.1. La partie ETL

8.1.1. Extraction des donn  es

Dans notre cas, les donn  es proviennent de la base de donn  e PostgreSQL de l'ERP, une base de donn  e, selon le nombre de modules install   dans l'ERP, peut avoir plus de 1 000 tables avec certaines tables comportant plusieurs centaines de champs avec des noms ambigus. Certains champs calcul  s dans l'ERP ne sont pas stock  s en base. Il faut alors d  cortiquer le code python pour reconstruire la valeur du champ, on peut vraiment parler d'extraire les bonnes informations de l'ERP. J'ai effectu   des sch  mas pour mieux voir o  

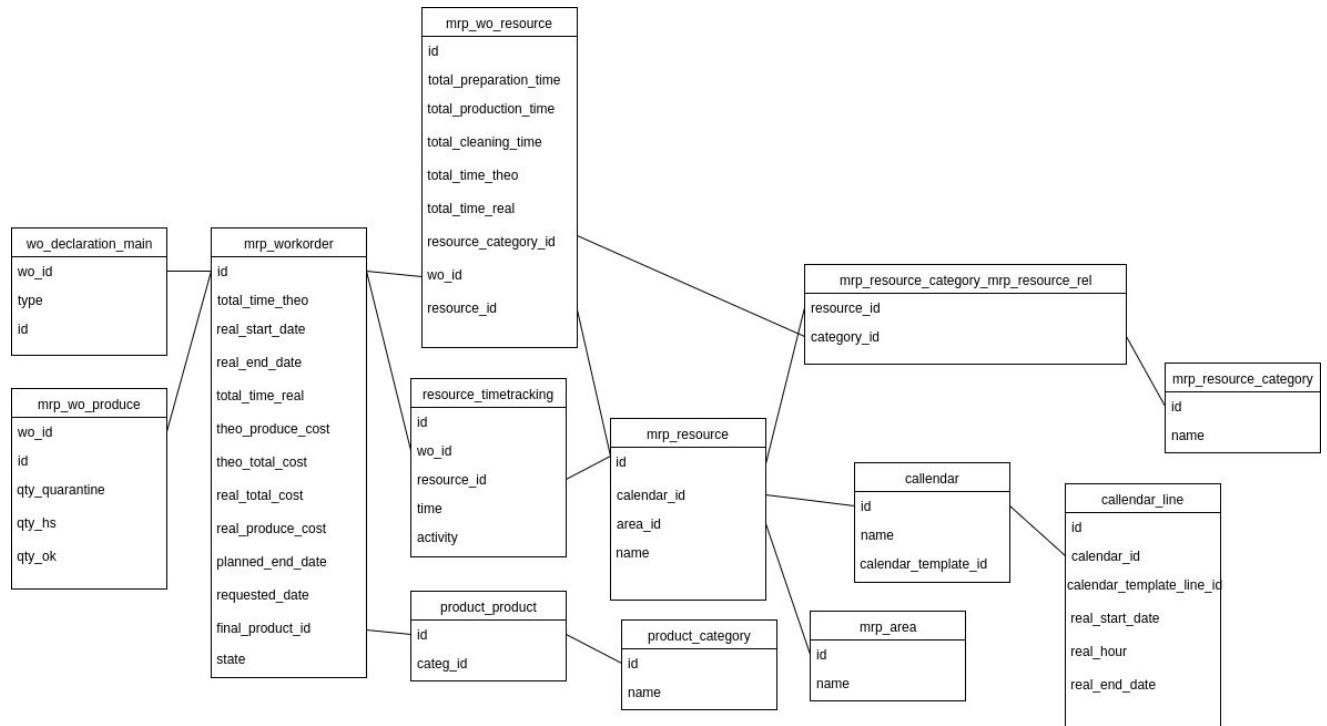
⁸ <https://eric.univ-lyon2.fr/~kaouiche/inf9002/mdx.html>

⁹

<https://support.pentaho.com/hc/en-us/articles/206021253-Best-Practices-Introduction-to-Mondrian-Performance-Tuning>

¹⁰ <http://community.pentaho.com/ctools/cdc/>

trouver les informations. Le schéma pour trouver les informations pour le cube des tableaux de bord de la production est visible ci-dessous :



8.1.2. Transformations

Dans notre cas, les transformations confiées à notre ETL sont souvent simples, comme la transformation de dates. Mais dans certains cas on effectue des requêtes SQL complexes pour retrouver la valeur de champ calculé dans le code Python de l'ERP qui ne sont pas stockés en base.

Dans l'ERP, les données peuvent être manquantes ou mal renseignées. La phase de traitement des données manquantes représente un moment important de l'ETL. Dans notre cas, on élimine les données qui sont mal formées dans la base de l'ERP, par exemple les id manquants. Dans certains cas, on crée un champ inconnu (par exemple « sans compte analytique », « sans client ») pour les axes d'analyses.

Lors de la transformation pour récupérer les coordonnées des adresses, j'ai fait une tentative de géocodage en envoyant l'adresse à un service web de géocodage. La première difficulté rencontrée fut que les services web limitent le nombre d'adresses à géocoder et rejettent une charge trop importante de demandes pour des raisons de sécurités et de performances. Les solutions payantes existent, mais leur défaut principal est bien sûr le fait qu'ils ne soient pas gratuits, j'ai donc fait des travaux de recherche pour comparer les

différents offres disponibles actuellement. Mon choix s'est porté sur locationIq¹¹ qui dans son offre gratuite permet le géocodage de 30 000 adresses par jour à une vitesse maximale de 15 requêtes par seconde, ce qui est amplement suffisant pour le plus gros client actuel de l'ERP. En faisant des tests sur divers bases des clients de l'ERP, je me suis rendu compte que près de 30 % des adresses n'étaient pas géocodées, ou géocodées seulement à l'échelle du pays à cause de fautes de frappes, ou du fait que le géocodage n'est pas encore assez développé dans tous les pays du monde contrairement au pays européens ou aux États-Unis. 30 % représentant un pourcentage trop élevé pour une bonne visualisation graphique, j'ai donc décidé de ne pas la rajouter à l'ETL final et m'arrêter à la granularité du pays pour l'axe d'analyse géographique.

8.1.3. Mise-à-jour de l'entrepôt de données

Un point à considérer est de savoir si les requêtes de BI sont effectuées sur les données de production en temps réel ou si on travaille sur des réPLICATIONS mises à jour régulièrement. Dans notre cas, un système de réPLICATION suffisamment automatique et régulier, est suffisant pour donner des indicateurs assez précis pour la BI tout en laissant l'ERP gérer de gros volumes de transactions.

Un autre point est la mise à jour des données, les données de l'ERP changent dans le temps, par exemple une facture peut passer de l'état « non payé » à « payé » ce qui aura un impact sur le chiffre d'affaires, l'utilisateur a la possibilité de supprimer des clients, modifier des données de production. Il est donc important que les données dans notre entrepôt reflètent l'image des données de l'ERP. On diffère ainsi de la datawharhouse « classique » dont on vient ajouter chaque jour les nouvelles entrées par le fait que les données vont être écrasées lors du lancement de l'ETL pour garder la cohérence entre les données de l'ERP et de la BI. Ainsi on lance toutes les 24 heures les transformations de l'ETL (un cron) qui vont venir mettre à jour les données de notre entrepôt. Pour que l'utilisateur ait la possibilité d'avoir les dernières mises-à-jour, on lui laisse notamment la possibilité d'exécuter le cron depuis l'interface web de l'ERP.

Lors de la mise-à-jour de l'entrepôt de données, des problèmes de plusieurs sortes peuvent survenir. On peut par exemple perdre la connexion avec le serveur si celui-ci tombe en panne. Dans ce cas on log les informations concernant l'erreur et on avertit l'administrateur de l'ERP par courriel pour que celui-ci soit mis au courant par un message clair de la source de l'erreur.

8.1.4. Conception des cubes

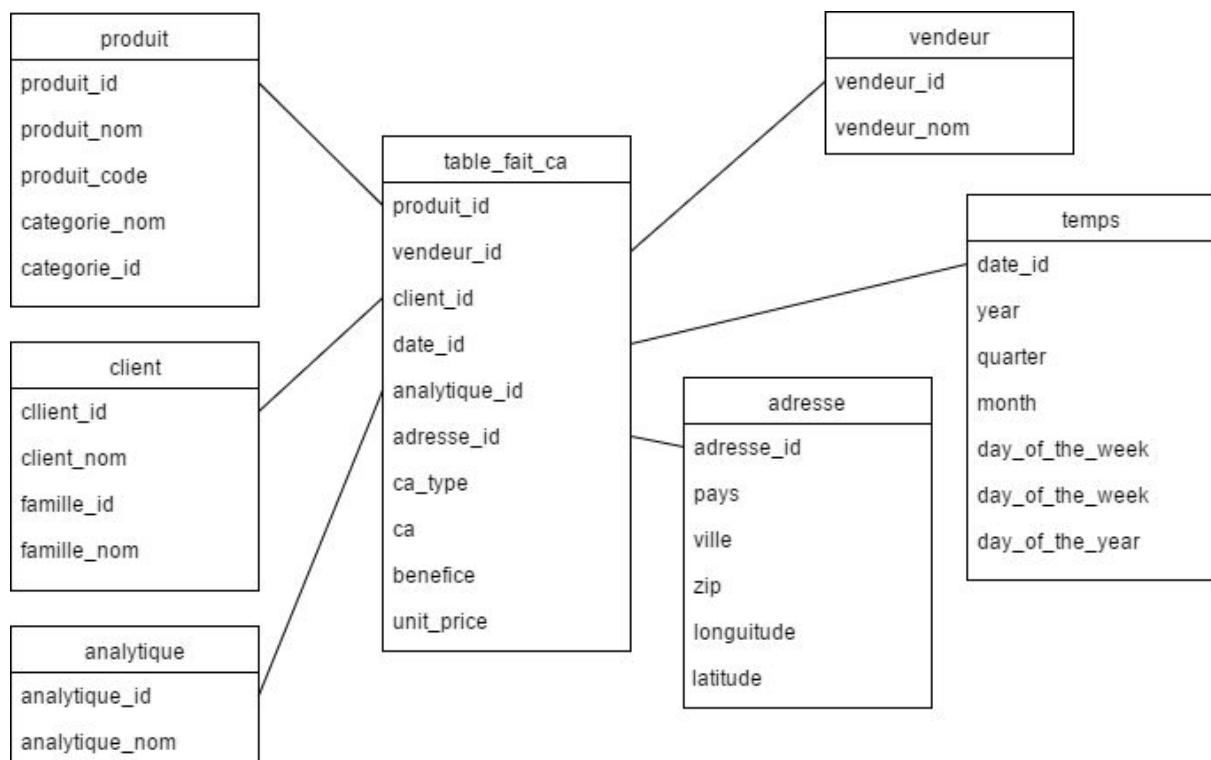
Comme toute modélisation de base de données, cette étape est cruciale dans la réussite d'un projet décisionnel. La modélisation doit être évolutive afin de facilement

¹¹ <http://locationiq.org/>

recevoir de nouveaux domaines fonctionnels si un client de l'ERP a des demandes particulières. J'ai donc décidé de suivre la modélisation en étoile, ainsi on peut facilement ajouter un axe d'analyse, ou une mesure à notre modèle.

Avant de commencer la conception des cubes pour l'analyse multidimensionnelle, il faut avoir une vue d'ensemble précise des données présents dans l'ERP, des informations qu'il est possible d'extraire, ou au contraire impossible d'extraire, car ils ne sont pas présents dans l'ERP. J'ai effectué plusieurs maquettes, en m'apercevant d'erreurs de conception avant d'avoir la version définitive. J'ai documenté chaque cube de donnée avec les schémas de ces derniers et les informations sur les tables, le nom des champs et les calculs à effectuer. Ainsi dans le futur, si un module de l'ERP change en base de donnée, le prochain développeur saura quelles données changer lors de la maintenance.

Table de fait pour les tableaux de bords de vente :



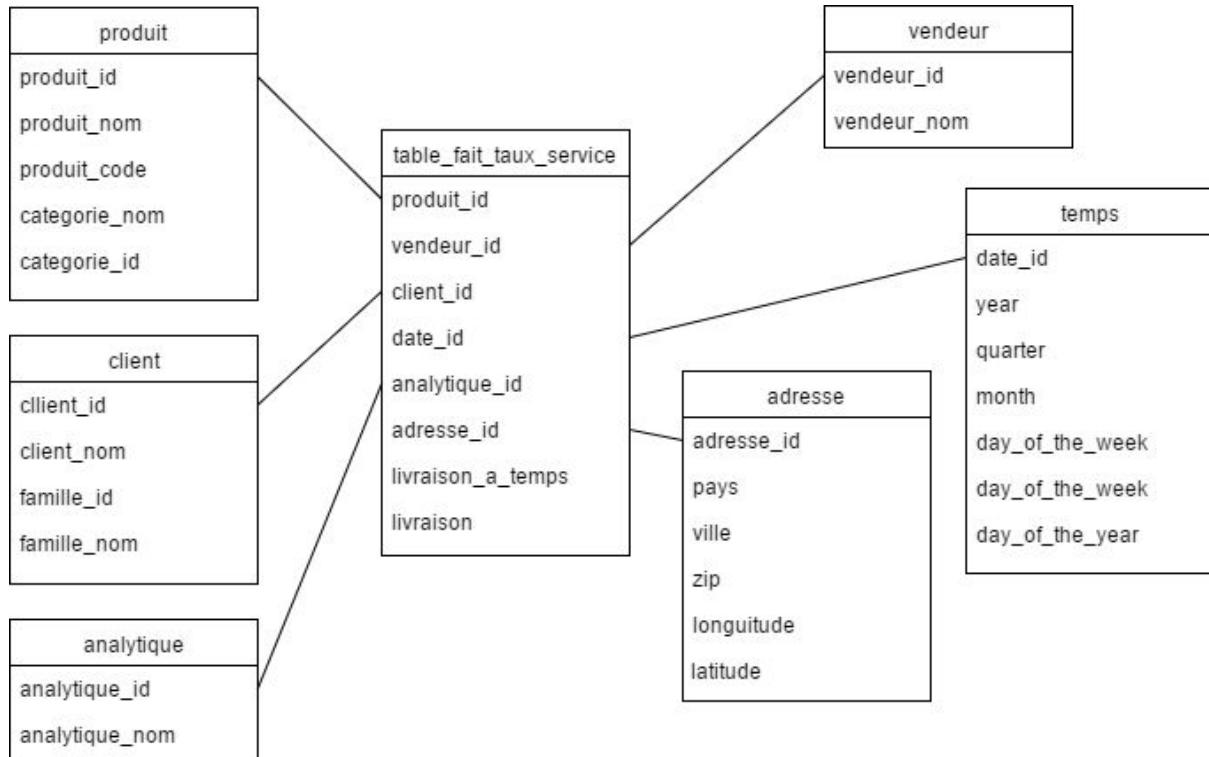
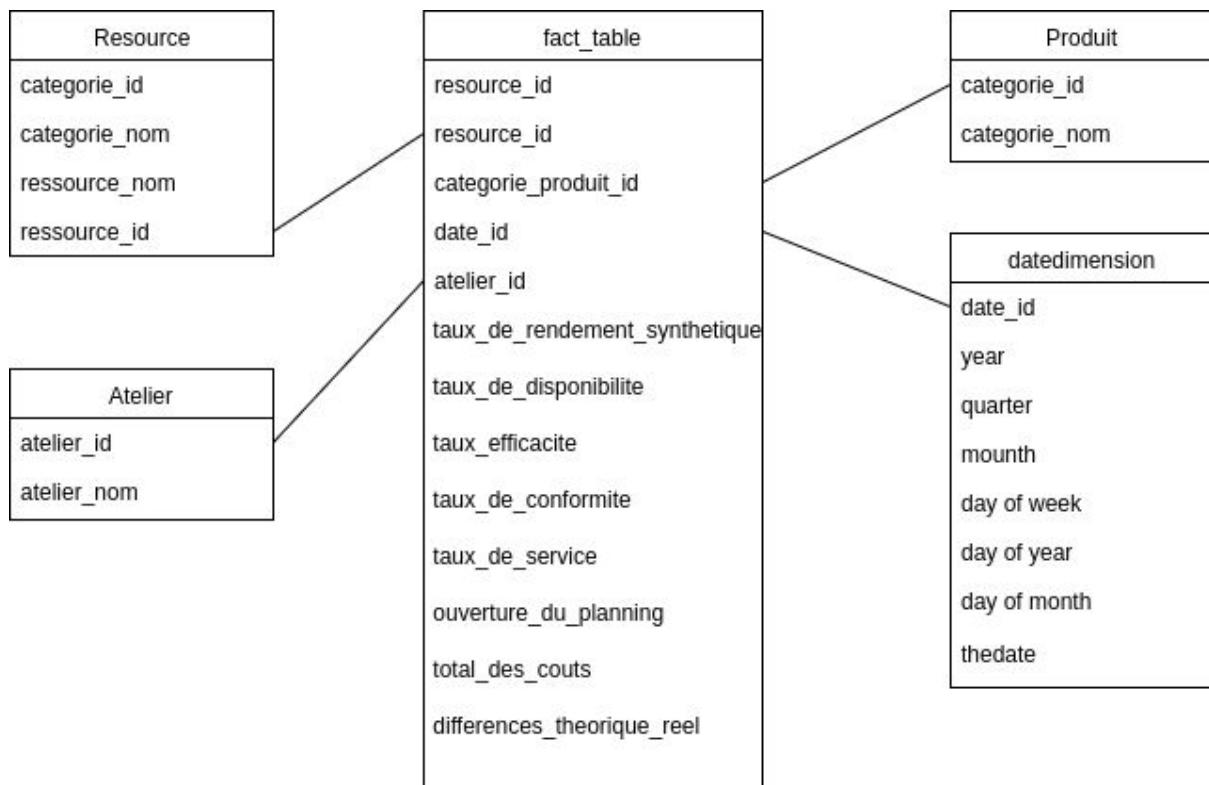


Table de fait pour les tableaux de bords de la production :



8.2. Les tableaux de bord

8.2.1. Internationalisation des tableaux de bord

L'ERP est disponible en français ou en anglais selon les propriétés du navigateur de l'utilisateur. Pour rester cohérent avec l'ERP, j'ai fait la traduction des tableaux de bord dans les deux langues en suivant la pratique de codage i18n¹² grâce au module de traduction disponible dans Pentaho. Chaque texte qui peut être traduit est placé dans une chaîne de ressource. Au moment du chargement de la page web, on vient charger le fichier correspondant à la langue de l'utilisateur s'il est disponible ou la langue par défaut (dans notre cas l'anglais) s'il ne l'est pas.

Pour une évolution future, si l'on souhaite traduire les tableaux de bord dans une autre langue, il suffit de rajouter le fichier correspondant à celle-ci avec les chaînes de ressources traduites sur le serveur Pentaho.

8.2.2. Conversion au format PDF des tableaux de bords

Lors de la convention du tableau de bord en pdf, plusieurs problèmes se posent :

- Il faut que l'utilisateur puisse imprimer le pdf, il lui faut donc un rendu à la bonne échelle.
- Le fait de pouvoir imprimer impose de faire les sauts de page au bon moment, pour ne pas casser un graphique en deux ou mettre le titre d'une section en fin de page.
- On souhaite cacher certains éléments au pdf, comme par exemple les boutons exporter, quitter, comparer.

J'ai effectué des travaux de recherche sur l'outil le plus approprié pour convertir les tableaux de bords en format pdf.

Des solutions ont été effectuées pour convertir les tableaux de bords Pentaho au format pdf :

- Installer un module dans le navigateur du client comme PDFCreator¹³ ou DoPDF¹⁴. L'installation d'un tel module demande de l'implication de la part de l'utilisateur et un même module est rarement disponible pour tous les navigateurs supportés par l'ERP, donc cette possibilité ne convient pas pour les tableaux de bords Pentaho.
- Créer un rapport avec Pentaho Report Designer et utiliser CGG (Community Graphics Generator) du projet Ctools de Pentaho pour créer les images des différents graphiques. Cette méthode nous permet d'avoir des PDF avec les graphiques précis au pixel près. Mais nos tableaux de bords étant dynamiques, le but étant est que l'utilisateur puisse cacher les parties du tableau de bord qui ne l'intéressent pas, voire agencer les éléments du tableau de bord à sa guise. Il faudrait créer autant de rapport que de possibilités de tableaux de bord, cette solution ne nous convient pas.

¹² [https://fr.wikipedia.org/wiki/Internationalisation_\(informatique\)](https://fr.wikipedia.org/wiki/Internationalisation_(informatique))

¹³ <http://www.pdfforge.org/>

¹⁴ <http://www.dopdf.com/>

- L'ERP utilise wkhtmltopdf¹⁵ pour les conversions PDF. En passant l'URL du tableau de bord au service wkhtmltopdf, celui-ci fabrique le pdf côté serveur et le renvoie au client. Le problème étant que les tableaux de bords sont dynamiques, on perdrait ainsi les modifications de l'utilisateur, cette solution ne nous convient donc pas.
- Utiliser PhantomJS¹⁶, un navigateur web sans interface graphique qui permet grâce à sa solution générique d'impression de pages web, de piloter l'impression PDF du tableau de bord à partir de son URL¹⁷. Les problèmes sont les mêmes que pour wkhtmltopdf. Pour palier ce problème dans phantomJS, on pourrait simuler les actions de l'utilisateur et ainsi obtenir l'état du tableau de bord de l'utilisateur mais cela serait long et complexe à mettre en place.

J'ai donc développé un module JavaScript qui fabrique un PDF de l'état actuel du tableau de bord en prenant des images des éléments visibles du DOM et les insère en faisant les sauts de page au bon moment dans un PDF que l'utilisateur peut ensuite télécharger.

8.2.3. Mise en place d'une Connection Pool

Un facteur important pour la performance de nos tableaux de bord est la liaison entre Pentaho et notre entrepôt de données. L'établissement de cette connexion a un temps non négligeable. La principale raison de cette latence est la nécessité d'établir une connexion TCP/IP entre Pentaho et notre base de données avec les contraintes naturelles que cela pose (lenteur du réseau, pare-feu, filtrages, quand l'entrepôt de données est sur une autre machine,...).

Pour pallier ce problème, on peut configurer une connection pool dans Pentaho.

Lors de la configuration d'une connexion pool, le nombre maximal de connexions ouvertes dans la connexion pool est importante. D'après mes recherches, j'ai trouvé une formule pour déterminer ce nombre, c'est cette formule qu'on utilise actuellement sur le serveur BI:

La formule est la suivante¹⁸:

nombre de connexions maximales = ((nombre de coeurs du serveur x 2) + le nombre de disques physiques du serveur)

8.3. Intégration de pentaho dans l'ERP

8.3.1. Authentification sur le serveur BI

Les contraintes d'intégration de Pentaho dans l'ERP sont les suivantes :

- Il faut une authentification: les données des tableaux de bords contiennent des informations sensibles pour l'entreprise, elles ne doivent en aucun cas être disponibles au grand public.
- L'authentification doit être faite de manière transparente pour l'utilisateur, pas d'écran de login, ou de nouvelle page à ouvrir dans le navigateur.

Pour s'authentifier au serveur Pentaho, plusieurs solutions existent :

¹⁵ <https://wkhtmltopdf.org/>

¹⁶ <http://phantomjs.org/>

¹⁷ <http://www.osbi.fr/exporter-en-pdf-des-dashboards-pentaho-cde-avec-phantomjs/>

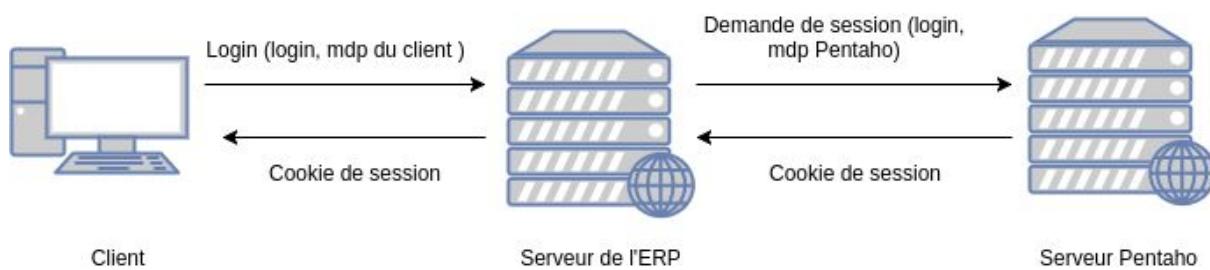
¹⁸ <https://github.com/brettwooldridge/HikariCP/wiki/About-Pool-Sizing#the-formula>

- Activer une option pour passer le nom d'utilisateur et le mot de passe dans l'URL, l'inconvénient de cette méthode est que quiconque écoute le trafic réseau serait en mesure de récupérer les identifiants
- L'authentification HTTP Basic¹⁹. Cette méthode est simple, mais également la moins sécurisée, car elle transmet le mot de passe codé en base 64 qui se décode aisément. Ce qui revient en termes de sécurité à passer le nom d'utilisateur et le mot de passe en claire dans l'URL.
- Le Central Authentication Service (CAS)²⁰ qui est un système d'authentification unique (SSO) pour le web. L'inconvénient est qu'il faut mettre en place un serveur CAS pour l'ERP chez tous les clients et changer le système d'authentification de l'ERP, ce qui demande un bon temps de développement.
- L'authentification par cookie, le serveur Pentaho reçoit une demande d'authentification pour créer une session avec les identifiants, puis il renvoie un cookie de session à celui qui a fait la demande. Tant que la session est valide des deux côtés, l'utilisateur est authentifié sur le serveur. J'ai choisi cette méthode pour l'authentification.

Implémentation de l'authentification :

Pour chaque utilisateur de l'ERP, on crée un mot de passe unique de 256 caractères avec la fonction os random de python, on crée les comptes correspondant sur le serveur Pentaho avec ce mot de passe et le login de l'utilisateur.

Lors du login de l'utilisateur sur l'ERP, l'utilisateur transmet ses identifiants au serveur de l'ERP, s'il est bien authentifié, le serveur de l'ERP envoie une demande de session au serveur Pentaho avec le login de l'utilisateur et son mot de passe Pentaho. Le serveur Pentaho si ce dernier est bien en état de marche renvoie un cookie de session au serveur de l'ERP, qui la transmet au client. Ce processus est illustré sur le schéma ci-dessous :



Il peut y avoir plusieurs erreurs ou exceptions lors du processus d'authentification du côté du serveur de l'ERP et de Pentaho qu'on gère en faisant remonter un message à l'utilisateur, à l'administrateur ou dans les logs du serveur de l'ERP en fonction du type de l'erreur.

¹⁹ https://fr.wikipedia.org/wiki/Authentification_HTTP

²⁰ https://fr.wikipedia.org/wiki/Central_Authentication_Service

9. Tests

A la fin de chaque cycle de développement, j'effectuais une suite de tests manuels pour couvrir les modifications effectuées lors du cycle. Quand le projet fut assez mûre pour afficher les premiers tableaux de bords fonctionnelles dans l'ERP, d'autres fonctionnels de la société ont pu tester le module. Grâce à leurs retours j'ai pu corriger diverses erreurs, améliorer la documentation et le guide d'installation.

C'est dans la phase de l'ETL que les tests prennent le plus de temps pour couvrir la large gamme de possibilités. Certains problèmes ne se sont montrés que sur des bases de données particulières de l'ERP lors de l'ETL, comme par exemple un champ avec la valeur null qui ne devait pas l'être ou un module non installé ou pas mis-à-jour dans l'ERP.

Dans le cadre de mon stage, j'ai été amené à découvrir le fonctionnement d'un module de test de l'ERP pour le migrer d'une ancienne version de l'ERP vers la version actuelle. Ce module a été développé pour faire du "behavior-driven development"²¹, une méthode agile de développement qui encourage la collaboration entre les développeurs, les responsables qualités et les intervenants non-techniques de l'ERP.

L'utilisateur écrit des suites de scénarios de tests en langage naturel. Il peut notamment générer une documentation avec des copies d'écran lors de l'exécution des scénarios.

Le module utilise selenium, un framework de test informatique développé en Java, cela nous permet de simuler le comportement d'un utilisateur dans plusieurs navigateurs et tester la qualité des données ainsi que l'affichage du résultat attendu décrits par les tests.

La migration demande une bonne compréhension de la méthodologie de test et de son fonctionnement, car il faut modifier le framework web de l'ERP et certaines règles de tests.

²¹ https://fr.wikipedia.org/wiki/Behavior-driven_development

10. Conclusion

Bilan

Durant mon stage, j'ai intégré un module de Business Intelligence dans l'ERP d'Open-prod en créant des tableaux de bord interactifs avec des indicateurs pertinents pour l'aide à la prise de décision. Chacun des tableaux de bord regroupe un secteur particulier de l'ERP. J'ai eu la chance de suivre ce projet depuis le commencement et je serai présent pour sa mise en production sur de nombreux serveurs prochainement.

J'ai réussi à intégrer les tableaux de bords de Pentaho dans l'ERP de manière transparent pour l'utilisateur. L'authentification au serveur BI de Pentaho depuis l'ERP se fait de manière automatique et sécurisé. J'ai automatisé le processus de l'ETL pour l'entrepôt de données servant pour la BI.

Concernant la performance, on a quelques secondes d'attente au chargement des tableaux de bords les plus complexes avec de grandes quantités de données pour le plus gros de nos clients. C'est un temps de réponse qui reste acceptable mais on espère le faire diminuer par la suite pour un temps de chargement quasi instantané des tableaux de bord.

Les tableaux de bords de la production et des achats sont encore à développer, il reste notamment un dernier cycle de développement sur le tableau de bord des performances de vente. Maîtrisant les technologies pour la création de ces tableaux de bord, leur développement sera plus rapide que pour les premiers, ainsi la fin du développement du module de BI est prévu pour fin juillet.

Compétences acquises lors du stage

Durant ce stage, j'ai pu découvrir et m'approprier plusieurs outils BI : le serveur BI de Pentaho, Kettle, l'outil servant pour l'ETL de Pentaho, CTools, un outil pour créer des tableaux de bords interactifs et le plugin Saiku qui effectue des analyses sur un cube OLAP.

J'ai notamment acquis des compétences sur l'ERP d'Open-prod. Mon maître de stage, M. Sylvain FEROT, responsable R&D de la société, nous quitte fin juin. Étant appelé à reprendre sa place pour le développement JavaScript sur l'ERP, j'ai passé le mois de mai et de juin en formation sur l'ERP d'Open-prod, il m'a transmis ses connaissances sur le framework web et l'ORM de l'ERP. J'ai participé à l'amélioration du framework, ainsi qu'au développement de modules sur des tablettes pour gérer la production dans l'ERP.

J'ai acquis des connaissances dans les langages Python, JavaScript, MDX et SQL.

L'avenir et l'évolutions de la BI au sein de l'ERP

Les outils de la BI Open Source évoluent rapidement. Il sera intéressant d'effectuer une veille technologique sur ces outils dans les prochaines années, notamment de suivre comment les autres ERPs vont intégrer le domaine du décisionnel dans leurs produits.

Dans le module de Business Intelligence développé au cours du stage, on propose des tableaux de bords déjà construit à nos clients. La prochaine étape pour une évolution future serait de permettre aux utilisateurs de construire leurs propres tableaux de bords en quelques cliques de souris. Le développement de tableaux de bords dans Pentaho à ce jour, demande des connaissances de développeur à l'utilisateur et n'est pas encore assez user-friendly pour le grand public. Le Ad-hoc reporting n'est pas encore disponible dans les outils gratuits de l'Open Source. Dans les solutions payantes de tels outils existent, néanmoins leurs utilisations restent complexe pour un utilisateur sans formation. Le projet Open Source "Saiku reporting"²² est fort promettant sur ce sujet, son évolution est à suivre, pour peut-être une intégration dans l'ERP dans les années à venir.

Ce stage chez la société Objectif-PI m'a beaucoup apporté sur le plan professionnel. J'espère pouvoir mettre en valeur les connaissances que j'ai acquises durant ces dernières années au profit de la société Objectif-PI en partageant un bout de chemin ensemble via le poste d'ingénieur en CDI qu'on me propose à l'issue de ce stage.

11. Bibliographie

Livre Blanc de Smile sur les ERP Open Source

Livre Blanc de Smile sur le décisionnel Open Source

Learning Pentaho CTools

12. Webographie

<http://www.osbi.fr/> Actualité de la Buisiness Inteligence dans l'Open Source

<https://www.pentaho.com/fr> Penthô, une suite de solution BI Open Source

<https://www.jaspersoft.com/fr> Jaspersoft, une suite de solution BI Open Source

https://www.odoo.com/fr_FR/ Odoo une ERP Open Source

<http://open-prod.com/ERP/fr/> Open-prod l'ERP d'Objectif-PI

²² <https://www.kickstarter.com/projects/2117053714/saiku-reporting-interactive-report-designer/>

13. Liste des illustrations

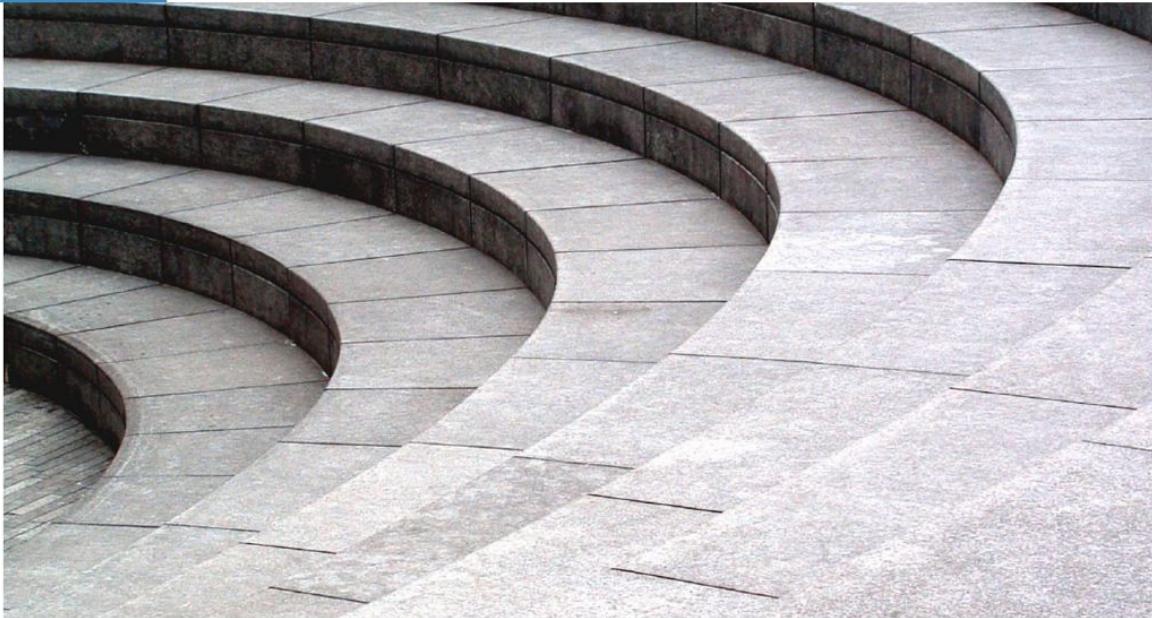
Graphique de force de vente présent dans l'ERP	12
Tableau croisé présent dans l'ERP	12
Evolution de l'intérêt pour la recherche Pentaho et Jaspersoft	14
Cycle de vie d'un composant du tableau de bord CTools, source https://www.pentaho.com	18
Schéma de la base de donnée de l'ERP avec les informations pour l'ETL des tableaux de bords de la production.	20
Table des faits pour les tableaux de bord des ventes	22 - 23
Table des faits pour les tableaux de bord de production	24
Authentification sur le serveur Pentaho en passant par le serveur de l'ERP.	26

14. Acronymes

- BI : Business Intelligence
- CA : Chiffre d’Affaire
- CAS : Central Authentication Service
- CDE : Community Dashboard Editor
- ERP : Enterprise Resource Planning
- ETL : Extract Transform Load
- JPEG : Joint Photographic Experts Group
- MDX: MultiDimensional Expressions
- OLAP : OnLine Analytical Processing
- OT : Ordre de Travail
- PDF : Portable Document Format
- PNG : Portable Network Graphics
- SQL : Structured Query Language
- SSO : Single Sign-On
- TCP : Transmission Control Protocol
- URL : Uniform Resource Locator
- XML : eXtensible Markup Language

Annexe I

Description des différents niveau de Business Intelligence intégrable dans un ERP.



Five Levels of Embedded BI: From Static Reports to Analytic Applications

Leveraging a BI platform with built-in multi-tenancy provides better tenant security and provisioning and increased branding opportunities. In addition, it creates one single, easy-to-manage system that supports multiple customers.

INTRODUCTION

The expanding role of data promises smarter operational applications that manage and automate better processes. These new, intelligent, analytic applications are transforming how other applications can use information to drive improved business performance and competitive advantage.

Most organizations derive value from their data through reports, dashboards, and analytic visualizations using stand-alone business intelligence (BI) and data warehouse tools. Yet only a small percentage of people actually use this style of BI due to user interface complexity, lack of current information, and data inaccuracy issues.

Operational applications already in use offer an opportunity to evolve from using static operational data exclusively to including interactive analytics that foster more powerful decision-making.

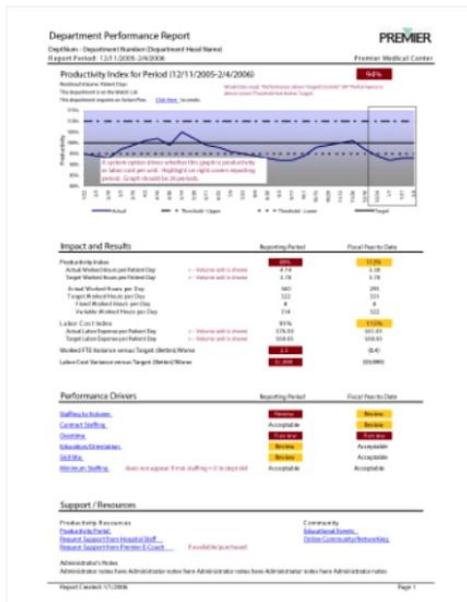
Using a fictitious business, Acme Fresh Delivery Foods, and use cases involving its IMS2 inventory management system, this ebook steps through five levels of increasing engagement and user value that can be achieved by embedding BI in to any application:

- Level 1: Static reporting using an embedded reporting library
- Level 2: Managed reporting with simple interactivity, scheduling, security, and distribution using a reporting server
- Level 3: Highly interactive reports and dashboards using a reporting server
- Level 4: Self-service ad hoc reports using a BI server
- Level 5: Self-service data exploration against a data mart using a BI server

The IMS2 inventory management system organizes product information, inventory units, location data, movement history, and bill of material information. It is used by a variety of personnel, including inventory clerks, warehouse managers, retail agents, shop floor managers, and executives.

LEVEL 1

STATIC REPORTING



while others are available to download as a Microsoft Excel worksheet. The reports provide a static view in time, typically coming from the live operational database within the application.

Application Developers Want an Easy Architectural Solution

Application developers have several options when determining how best to provide a static report for their operational application. They can build their own tool, embed an open source reporting engine, or buy a commercial solution. They can look at several architectural options to seamlessly enable reporting without disrupting the architecture of the application. Limiting the impact on the application architecture usually results in a limited reporting solution for end users.

Considerations

An embedded reporting library allows a user to either run a report on demand or request the application run the report in the background and store it. The library's pre-built reports are designed by a developer who defines the report layout and export formats (PDF, XLS, HTML). Each report must be designed to prevent significant performance hits to the application. The reporting library is often built inside the application in the same programming language. Additional code is typically written to manage access, security, scheduling, and report storage.

Embedded BI Components to Look For

- Reporting library for services such as report compilation, layout, and export format
- Desktop-based report designer for report creation

Example:

Joel is an inventory clerk at Acme Fresh Delivery Foods retail store in Phoenix. He wants to know how many batteries his distributor provided last month and how many are planned for the rest of the year, presented by week in a table and bar chart. He wants to run the report against the latest, most up-to-the-minute data. The application administrator, Kelly, implemented the report as a menu option in the IMS2 application; it is output in PDF format.

End Users Just Want a Report

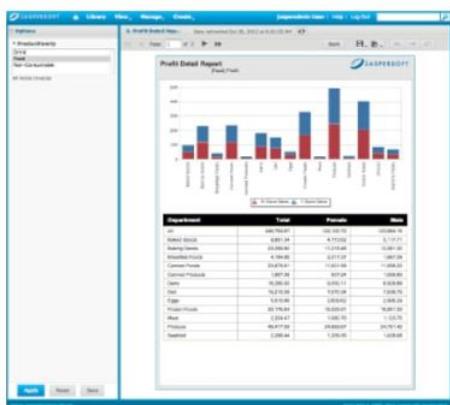
When users want to view data from their operational application, it's typically provided through a static report. Some reports are formatted and ready to print,

Limitations

- Static point-in-time information: Embedded reports typically present a limited timeframe of information based on a lack of historical data stored within the application's database. This means reports can't show trends. Additionally, the report is usually static and doesn't allow users to drill from summary data to underlying details to gain further insight.
- Unfulfilled Report Requests: Each embedded report is designed by an application developer, which means assumptions are made about how and what to present to the end user. New report requests can only be addressed by the developer, which means custom report requests go unfulfilled or are delivered according to the developer's availability.
- Impact to Application Performance: If the application does not provide report scheduling and a report repository, the same report query could be run repeatedly by different users at the same time, impacting the application's performance. Additionally, the compilation and layout of each report takes computational resources.
- Developer Effort: With each new report, the developer must consider the performance impact on the operational application and any security implications. As end users request new views to the application data, the developer must balance new report requests against new feature enhancements.

LEVEL 2

MANAGED INTERACTIVE REPORTS



Example:

Janet is a manager for Acme Fresh Delivery Foods' western regional stores. She wants to know how many total shipments were delivered last month per distributor, how many items are in stock, and how many will be shipped next month for each store in her region. She then wants each report to run nightly and be sent via email to each store manager. Paul, the manager of the Santa Fe store, has requested an inventory aging field for his store's report. Kelly, still responsible for maintaining and extending

IMS2, knows the current level of embedding isn't up to the task.

Managed Application Reporting for Better Business Performance

The success of an operational application often leads to new business requirements and new challenges for the group managing the application. Managed reports help drive business performance with information sharing and pre-defined key performance indicators (KPIs).

Kelly knows she can't support Janet's and Paul's requirements with an out-of-the-box Level 1 reporting solution. She would need to build a scheduler and report distribution service as a customization. Additionally, Kelly needs to build a handful of new custom reports, either by extracting the data from the application and building reports with another tool, or by requesting an enhancement of the IMS2 vendor. As is often the case, she is also responsible for other tasks, such as maintaining other applications, which results in delays to the fulfillment of Paul's and Janet's requests.

Considerations

Automated delivery of pre-built and custom reports is an expected feature in most enterprise operational applications. Application end-user engagement and efficiency increases when the data needed is easily available and used for day-to-day planning and decision-making. If the application gets in the way of progress, the possibility of new solutions or competitors rises. Level 2 reporting requirements can be met by complementing the reporting library with a report scheduler, reporting repository, report distribution service, role-based security, and a report designer for new report requests.

The application can provide these services natively with a fair amount of development effort, or the services can be provided by an integrated, packaged BI solution. With these services enabled, the embedded BI solution can provide more reports to more users.

Embedded BI Components to Look For

- BI server for data security, metadata layer, dashboard framework, and reporting services (scheduling, distribution, and organization)
- Desktop-based report designer for highly complex reports
- Customizable UI framework for seamless branding and integration with the operational application

Limitations

A reporting server can improve information delivery and engagement within a company through scheduled, interactive reports. It can also improve operational application performance by offloading report compilation to a separate reporting server. But some limitations restrict the application's ability to address the changing dynamics of a business:

- Simplistic Data Security: A simple report object model such as a Level 2 reporting solution does not provide query (SQL) level security. A reporting repository does not inspect report queries, so the report developer or administrator must build the security attributes within the report and assign these attributes to the report object manually.
- Limited Availability of Custom Reports: New custom reports require the expertise of a professional report developer due to the complexity of the underlying data source, security models, and report formatting requirements. Most organizations don't have professional report developers, so the flow of new reports to business users can be slow.
- No Dashboard Reports: Once the basic reporting requirements are met, the application's users soon request more complex features, such as executive views. This type of reporting is enabled through dashboards, which provide at-a-glance summaries of critical performance indicators. Most dashboards support the ability to drill down from a summary view into underlying details for further inspection.

LEVEL 3

HIGHLY INTERACTIVE REPORTS AND DASHBOARDS



Example:

Steve is the head of inventory operations for all Acme Fresh Delivery Foods stores. He wants to see up-to-the minute metrics for key inventory and point-of-sale (POS) performance indicators, presented in a single easy-to-read dashboard. He wants his reports to be interactive and provide drill down to detailed data,

filters, and easy-to-see flags associated with metric outliers. These metrics from IMS2 are combined with another POS application to provide a central business performance dashboard. Kelly realizes once again that the current solution isn't going to be adequate for this new level of service. She will have to take it to the next level to really wow Steve.

ADDRESSING NEW USER PROFILES WITH APPLICATION DASHBOARDS

While the need for reports mentioned in the first two phases provides detailed application information for daily tactical decisions by line managers and shop floor users, it may not suit an executive or line of business manager. Executives usually don't use operational applications daily, but do rely on weekly, daily, or hourly snapshots of business performance. This view of information is usually presented in an easy to consume interactive dashboard.

A performance dashboard measures short- and long-term trends with quick access to underlying details to help managers react tactically or strategically to their business needs. Dashboards can also feed from several different application sources to present a holistic view of the business.

Considerations

Level 1 and 2 reporting can't effectively present an interactive dashboard. Dashboards are a collection of small reports, called dashlets, that are assembled on a single canvas, often with interactive controls that allow the user to change the view of data by time, location, or other parameter. The framework to control these integrated dashlets requires an orchestration layer typically managed by a metadata layer within the reporting server environment. To improve utility and draw decision-makers to the dashboard, the overall layout and design requires compelling elements like interactive charts and drill-down-enabled dashlets so users can quickly explore and better understand the business activity underlying the dashboard's information.

Embedded BI Components to Look For

- BI server for data security, metadata layer, dashboard framework, and reporting services (scheduling, distribution, organization)
- Desktop based report designer for highly complex reports or dashlets
- Customizable UI framework for seamless branding and integration with the application

Limitations

Level 3 embedded BI allows new user profiles to use the data stored within the IMS2 application. Dashboards can drive new strategies, improved decisions, and better planning. However, Level 3 does not alleviate the continuing requests for custom reports by other types of users. Success at this level often surfaces these new requirements:

- Lack of Custom Reports: New custom reports require the expertise of a professional report developer due to the complexity of the underlying data source, security models, and report formatting requirements. Most organizations have few to no professional report developers, a situation that can impede response to business users' requests for new reports. The ideal solution is a report design tool that less technical users can use to build their own reports without relying on IT or skilled report developers.
- Lack of Data Exploration and Analysis: Dashboards help visualize complex processes in easy-to-understand terms. However, a compelling dashboard inspires user curiosity and the desire to learn more about their data and why a metric is under- or over-performing. The answers to these questions are often beyond the scope of the dashboard and its underlying detailed reports. Supporting these deeper, spur-of-the-moment questions requires exposing the data through an interface that end users can interact with. Data exploration often requires queries comparing various products, locations, and times. Users want to look at the data in different dimensions to find trends or outliers.

LEVEL 4

SELF-SERVICE REPORTING FOR OPERATIONAL APPLICATIONS

The screenshot shows a software interface titled "TIBCO Jaspersoft". The main window displays a grid titled "Product Results by Store Type". The columns are "Product Name", "Brand", "Description", "Category", "Store Type", "Sales", and "Profit". The data includes various products like "Lemonade", "Orangeade", "Milk", "Yogurt", etc., across different store types such as "Retailer", "Wholesaler", "Food Court", "Gas Bar", "Bakery", "Grocery", "Drugstore", "Supermarket", "Convenience Store", "Gas Station", "Food Truck", and "Mobile". A sidebar on the left lists categories like "Data Sources", "Reports", "Dashboards", "Visualizations", "Jobs", "Logs", and "Metrics". On the right, there are several filter panels: "A Country", "A Manager", "A City", and "A Date Range". Below the main grid, there's a "Report Filter Expression" section with dropdowns for "Report", "Type", "Value", and "Match".

metrics. The current static reports provided by the IMS2 system don't provide the flexibility to generate these reports, forcing Paul to request new reports from Kelly every month or so. Kelly is getting similar requests for custom reports from other employees and can't keep up with demand.

Considerations

When Acme Fresh Delivery Foods knowledge workers want their own custom reports that are not part of the packaged operational application, there are two options:

Option 1:

Provide Paul or Kelly direct database access to the operational application schema, allowing them to pull a CSV dump of the data to their local machine, or give them a report design tool for more advanced requirements.

The downside to this option are the training costs and resources required to teach non-technical users how to use powerful, full-featured report design tools to build each report manually, along with the associated lag time. There is added risk to application performance due to the uncontrolled queries that this approach introduces to the environment. For larger implementations, there is added security risk in providing general access to the database, as well as additional training costs for the larger group.

Option 2:

Provide Paul an easy-to-use report design environment for his ad hoc reporting and analysis needs. Kelly can define an easy-to-understand semantic layer that sits on top of the application database. This layer allows non-technical users to understand the column names and the data while providing a security access model for the underlying database. The metadata can be designed against the operational database or a specialized data mart.

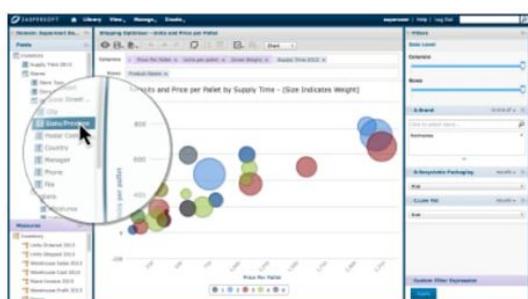
With these elements, and a graphical drag-and-drop report designer, knowledge workers like Paul can now build their own reports on demand, without requesting support from Kelly. In addition, an in-memory query cache can further reduce the impact to the application database, enabling custom reports that don't impact application performance. When integrating a self-service BI environment with another application, there's a sizeable risk of losing the consistent look and feel of the original application. Select BI tools able to change their UI to blend seamlessly with your application.

Limitations

- **Consistency of Application Look and Feel:** The look and feel of an application is important for software vendors and end users. Application developers who integrate an existing off-the-shelf BI platform should look for tools that allow easy customization that provides a consistent look and feel.
- By definition, self-service reporting tools should be easy to use for a non-technical user like Paul. They should require little training to build a self-service report, the reality is that many of the BI platforms offered today do not meet this requirement.
- If the application database contains lots of data, the amount of data stored within an in-memory engine is limited to the amount of memory available on the computer hosting the BI server. If the application database is large, it's often better to configure a specialized analytic application database that leverages columnar storage and data compression.

Embedded BI Components to Look For

- BI server for data security, metadata layer, web-based easy-to-use report designer, dashboard framework, reporting services (scheduling, distribution, organization)
- Desktop-based report designer for highly formatted reports
- Customizable UI framework for seamless branding and integration with the operational application

LEVEL 5**SELF-SERVICE DATA EXPLORATION FOR DEEPER INSIGHT****Example:**

Susan is a product line manager for Acme Fresh Delivery Foods' headquarters. She wants to understand why margin costs are dropping for deliveries from the northwest stores compared to those in the Southwest. She would like to explore a variety of data and dimensions including wholesale unit costs, retail price, inventory aging, shipping costs, and product promotion data.

SOLUTION GUIDE | 8

She plans to compare products by store and date (month and year), including pulling in data from Acme Fresh Delivery Foods' product promotions system. She'll be looking for trends over time, possibly caused by seasonal store patterns and additional expenses on the supply side. To address this new need, Kelly considers these options:

- Develop custom reports that collect the data Susan needs
- Give Susan data exploration tools that let her intuitively find the data she needs

Considerations

One way Kelly can get answers to Susan's questions is to create several more custom reports. However, it's time consuming to manually develop, execute, and review each report. It's also inefficient to access the data this way, as it can degrade the performance of the operational application itself.

Different options can help solve this problem. A pre-built view into the underlying data that is structured for the sole purpose of analysis lets data analysts quickly inspect large datasets and perform complex queries over periods of time, which is not easily done with a transactional database.

Business analytics through data exploration can be a source of competitive advantage for a business, and including it into operational applications like IMS2 can mark a strategic move for Acme Fresh Delivery Foods. While self-service reports allow users to create their own tabular reports, data exploration allows aggregations across multiple dimensions (for example, average sales by product, region, and time period). The visualizations created are usually charts or crosstabs and allow easy comparative and time-based analysis (for example, drill-down, slice-and-dice, pivot, and filter).

Embedded BI Components to Look For

- Data integration tools for extracting, transforming, and loading data from the operational application database to a data warehouse or data mart
- Data warehouse or data mart for efficient processing of data for analytic purposes
- Either an in-memory engine, push-down aggregations, or a traditional OLAP engine, for performance and analytic processing
- BI server for data security, metadata layer, data visualization, web-based easy-to-use report designer, dashboard framework, and reporting services
- Desktop-based report designer for highly formatted reports
- Customizable UI framework for seamless branding and integration with the operational application

Limitations

While providing greater access to data presents a host of benefits to organizations, including insight into new revenue categories, improvements in operational processes, competitive advantages, and more, the cost to the application developer and the application administrator can be high. A few specific challenges for an embedded analytics solution are:

- Architectural Complexity: To properly deliver responsive and powerful advanced analytics, the environment must include additional services, including a database for analytic queries, data integration software, and metadata for aggregation definitions.
- Application Maintenance: With added software comes added maintenance for administrators. A common request by data analysts is to provide a new view into the data warehouse that requires a new data integration job and new query definitions.

CONCLUSION

There are a number of options available when electing to enhance an application with Business Intelligence. It's important to understand the benefits and user requirements associated with each level of capability. Acme Fresh Delivery Foods progressed through all five levels of embedded BI. Recognizing the value of the data within the IMS2 application, Kelly can now give Steve, Janet, Paul, and Susan the data and tools they need to make decisions when and how they need to. Each level of BI brought the Acme Fresh Delivery Foods team closer to realizing the value of their data and delivering a clearer understanding of their business. Embedded BI can make your applications and customers more knowledgeable—and it can give you the opportunity to deliver greater value and end-user engagement for operational applications.

Once you determine how to proceed, consider whether to build your own BI framework or buy a packaged solution. Consider these questions:

- 1 Do you have the time and resources to build and maintain an embedded BI solution for your application?
- 2 Do you have the expertise in BI to address emerging trends such as mobile and social capabilities?
- 3 Do you want your embedded BI to be delivered in a seamless unified user interface?
- 4 Will your application be delivered as a SaaS and/or as on premise?

LEARN MORE

TIBCO Jaspersoft® empowers millions of people every day to make faster decisions by bringing them timely, actionable data inside their apps and business processes. Jaspersoft is an embeddable, cost-effective reporting and analytics platform that allows anyone to quickly get the answers they need. It scales architecturally and economically to reach everyone. Thanks to a community that is hundreds of thousands strong, Jaspersoft commercial open source software has been downloaded millions of times and used to create the intelligence inside hundreds of thousands of apps and business processes. For more information, visit <http://www.jaspersoft.com> and <http://community.jaspersoft.com>.



Global Headquarters
3307 Hillview Avenue
Palo Alto, CA 94304
+1 650-846-1000 TEL
+1 800-420-8450
+1 650-846-1005 FAX
www.tibco.com

TIBCO Software empowers executives, developers, and business users with Fast Data solutions that make the right data available in real time for faster answers, better decisions, and smarter action. Over the past 15 years, thousands of businesses across the globe have relied on TIBCO technology to integrate their applications and ecosystems, analyze their data, and create real-time solutions. Learn how TIBCO turns data—big or small—into differentiation at www.tibco.com.

©2015, TIBCO Software Inc. All rights reserved. TIBCO, the TIBCO logo, TIBCO Software, and Jaspersoft are trademarks or registered trademarks of TIBCO Software Inc., or its subsidiaries in the United States and/or other countries. All other product and company names and marks in this document are the property of their respective owners and mentioned for identification purposes only.
09/21/15

Annexe II

Schéma, légendes et indicateurs des tableaux de bord

Les Tableaux de bord :

Performances de vente - bilan général
Performances de ventes - comparatif
Force de vente - vue d'ensemble
Force de vente - comparatif
Production - Bilan général
Production - Comparatif
Production - Rendements
Production - Disponibilité
Production - Autres (Coûts, Rebus, efficacité et taux de service)

Performance de ventes - Bilan général

Indicateurs

Indicateur	Formule de calcul	Axes d'analyses
Chiffre d'affaires réalisé (CA facturé)	Période t = Lignes de factures HT de la période – avoirs de la période	<ul style="list-style-type: none"> ● Catégorie Client/ Client ● Catégorie produit/produit ● Zone géographique ● Comptes analytiques ● Vendeur ● Périodes (mois quartier / année)
Chiffre d'affaires à venir (CA prévus non facturé)	<p>Pour chaque ligne de vente dont la date d'arrivée est sur la période t</p> <p>Période t = les montants des lignes de ventes validées – les lignes de factures clients HT liées à ces lignes de vente</p> <p>Si la différence est négative, on mettra le résultat à zéros</p>	<ul style="list-style-type: none"> ● Catégorie Client/ Client ● Catégorie produit/produit ● Zone géographique ● Comptes analytiques ● Vendeur ● Périodes (mois quartier / année)
Chiffre d'affaires prévisionnel (CRM)	Période t = le CA des opportunités x % de réussite - Catégorie Client/ Client	<ul style="list-style-type: none"> ● Catégorie Client/ Client ● Zone géographique ● Vendeur ● Périodes (mois quartier / année)
Taux de rentabilité	<p>Période t = (Prix de vente – prix de revient) / (prix de vente)</p> <p>Prix de vente = prix unitaire de la ligne de facture</p> <p>Prix de revient = Si le produit à une méthode de fourniture par défaut « Acheter », prendre la moyenne des 5 derniers achats (si produit acheté) sinon s'il a une méthode de fourniture par défaut « Production », prendre le prix de revient de la fiche produit (si produit fabriqué)</p>	<ul style="list-style-type: none"> ● Catégorie Client/ Client ● Catégorie produit/produit ● Zone géographique ● Comptes analytiques ● Vendeur ● Périodes (mois quartier / année)
Taux de service	Période t = Nb de livraison à temps/ Nb de livraison total	<ul style="list-style-type: none"> ● Catégorie Client/ Client ● Catégorie produit/produit ● Zone géographique ● Comptes analytiques ● Vendeur ● Périodes (mois quartier / année)

Listing des commandes en cours	Période t = les lignes de ventes non terminées (date d'arrivée < à la fin de la période)	<ul style="list-style-type: none"> • Catégorie Client/ Client • Zone géographique • Vendeur • Périodes (mois quartier / année)
Encours de factures dépassées	Formule = Montant des factures non payées et dont la date d'échéance est dépassée du jour actuel	<ul style="list-style-type: none"> • Catégorie Client/ Client • Zone géographique • Vendeur • Catégorie produit/produit
Encours de factures non payées	Formule = Montant des factures non payées sur la période du jour actuel	<ul style="list-style-type: none"> • Catégorie Client/ Client • Zone géographique • Vendeur • Catégorie produit/produit

Légende :

1. Choix entre deux types de BI vente : Performance de vente et Force de vente
2. Liste des filtres actionnés (Peuvent être supprimés par la croix)
3. Différents filtres pour affiner les indicateurs
4. Bouton permettant d'ouvrir un onglet afin de réaliser des comparatifs sur les différents indicateurs
5. Chacun de ces graphiques sera dynamique : En cliquant sur une de leurs catégories, cela entraînera la création du filtre correspondant à cette catégorie sur l'indicateur correspondant ainsi que sur l'ensemble des autres indicateurs de la page.
Pour revenir à l'état initial, il suffira de supprimer le filtre dans la liste filtre (cf 1)
6. Chacun de ces modules peuvent être gérés à la convenance de l'utilisateur : disposition sur la page, suppression....
7. Chacun des onglets peuvent être gérés à la convenance de l'utilisateur : titre, disposition sur la page, taille....
8. Ces indicateurs ne devront plus être affichés si un filtre sur une période, ne prenant pas en compte le jour actuel, est réalisé OU Laisser les indicateurs à zéro (ça peut être considéré comme une information)
9. Pour la répartition des familles client (vs des catégories produit), en cliquant sur une famille de client (vs une catégorie produit) à partir du graphique, cela modifiera le graphique pour faire apparaître les clients (vs les produits) correspondant : voir l'onglet Sales performances – client et produit
10. Ces trois filtres permettent d'avoir 3 types de CA :
 - Le CA facturé (période passée jusqu'à aujourd'hui)
 - CA non facturé mais validé (futur)
 - CA prévisionnel lié à la CRM (futur)
Chacune des cases à cocher fera évoluer le graphique du dessous
11. L'idée est d'afficher la période précédente à laquelle fait référence le pourcentage. Par exemple, si l'ensemble de la page est sur l'année 2017, le pourcentage correspondra

au ratio entre l'année 2016 et l'année 2017, il faudra donc le marquer entre parenthèse : *(Par rapport à l'année 2016)*

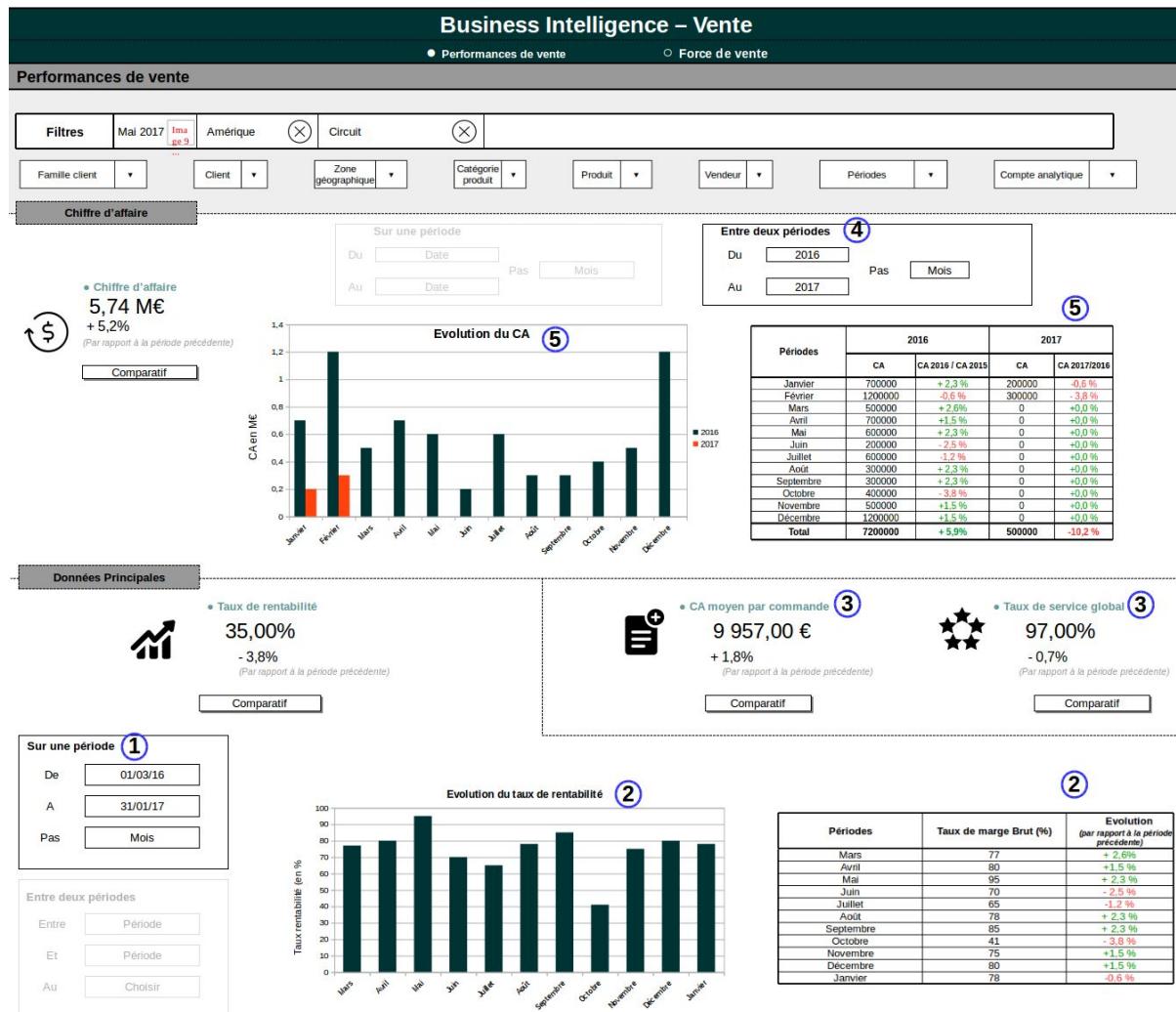


Performances de ventes - partie comparatif

Légende :

1. Le comparatif « sur une période » correspond à l'évolution de l'indicateur sur cette période suivant un pas donné (mois, trimestre (quarter), année...)
2. On ne compare ici que l'évolution de l'indicateur entre les différents « pas » de la période.
Par exemple, ci-contre on peut comparer l'évolution du chiffre d'affaires entre Mars 2016 et Avril 2016 (Le pas correspond ici à un mois). Par exemple, ci-contre on peut comparer l'évolution du chiffre d'affaires entre Mars 2016 et Avril 2016 (Le pas correspond ici à un mois).
3. Ces trois indicateurs pourront être rendus plus « transparent » pour avoir une meilleure visibilité de la page
4. Le comparatif « entre deux périodes » correspond à l'évolution de l'indicateur de la première période renseignée à la deuxième période renseignée. Le « Au » permet d'affiner la comparaison. Par exemple, on veut pouvoir comparer deux années non pas sur leur total mais mois par mois, ou trimestre par trimestre. Dans ce cas, on renseignera cette information dans « Au » (mois, trimestre, année...).
5. Dans cet exemple, on compare donc l'évolution de l'indicateur entre les années 2016 et 2017 au mois, c'est-à-dire que l'on va comparer janvier 2016 avec janvier 2017, février 2016 avec février 2017...

Remarques : Les périodes sélectionnées dans cet onglet « comparatif » ne devront pas créer un filtre pour l'ensemble des autres indicateurs de la page.



Forces de vente - Bilan général

Détail des indicateurs

Indicateur	Formule de calcul	Axes d'analyses
Ventes validées	Période t = Montant HT des ventes validées sur la période	<ul style="list-style-type: none"> • Vendeur • Périodes (mois quartier / année)
Ventes prévisionnelles et objectif	Il suffit de récupérer les champs « forecast » et « target » qui seront rentrés par les utilisateurs	<ul style="list-style-type: none"> • Vendeur • Périodes (mois quartier / année)

Légende

1. Liste des filtres actionnés (Peuvent être supprimés par la croix)
2. Différents filtres pour affiner les indicateurs.
On ne pourra filtrer que sur les périodes (Mois, trimestre (quarter), année...) du passé et du présent
3. Cette fonction permet de rajouter au bas de la page, les indicateurs (CA prévisionnel, objectif et validé) de la période sélectionnée de l'année passée. Voir l'onglet Force de vente comparatif
4. Menu déroulant pour avoir plus de détails sur la répartition des ventes

Remarques : Chacun de ces graphiques sera dynamique : En cliquant sur une de leurs catégories, cela entraînera la création du filtre correspondant à cette catégorie sur l'indicateur qui lui correspond ainsi que sur l'ensemble des autres indicateurs de la page. Pour revenir à l'état initial, il suffira de supprimer le filtre dans la liste filtre.

Bouton pour réduire la page et revenir à l'état initial.



Production - Bilan général

Détail des indicateurs

Indicateur	Formule de calcul	Axes d'analyses
Taux de Rendement synthétique	Periode t = taux de disponibilité * taux d'efficacité* taux de conformité	<ul style="list-style-type: none"> ● Catégorie ● Ressource/Ressource ● Période ● Atelier ● Catégorie Produit
Taux de disponibilité	Periode t = temps total réel de production (temps de production + temps de préparation + temps de nettoyage) / temps d'ouverture du calendrier	<ul style="list-style-type: none"> ● Catégorie ● Ressource/Ressource ● Période ● Atelier ● Catégorie Produit
Taux d'efficacité	Periode t = Temps total théorique de production (temps de production + temps de nettoyage + temps de préparation) / Temps total réel de production (temps de production + temps de nettoyage + temps de préparation)	<ul style="list-style-type: none"> ● Catégorie ● Ressource/Ressource ● Période ● Atelier ● Catégorie Produit
Taux de conformité	Periode t = Nombre de pièces conformes(total de la production – rebus - quarantaine)/ Nombre de pièces effectivement produites	<ul style="list-style-type: none"> ● Catégorie ● Ressource/Ressource ● Période ● Atelier ● Catégorie Produit
Taux de service	OT = ordre de travail Taux de service interne Periode t = Nombre d'OT dont la date réalisée ≤ date planifiée / Nb d'OT total Taux de service client Période t = Nombre d'OT dont la date réalisée ≤ date demandée / Nb d'OT total	<ul style="list-style-type: none"> ● Catégorie ● Ressource/Ressource ● Période ● Atelier ● Catégorie Produit
Temps d'ouverture du planning	Periode t = temps d'ouverture du planning de la ressource	<ul style="list-style-type: none"> ● Catégorie ● Ressource/Ressource ● Période ● Atelier ● Catégorie Produit
Total des coûts (théoriques ou réels)	Période t = somme des coûts réels ou théoriques de chaque OT de la période (c'est-à-dire les OT dont la date de fin réelle est sur la période)	<ul style="list-style-type: none"> ● Catégorie ● Ressource/Ressource ● Période ● Atelier ● Catégorie Produit

Différences coûts théoriques et réels	Période t = Différence entre la somme des coûts réels et la somme des coûts théoriques de chaque OT de la période (c'est-à-dire les OT dont la date de fin réelle est sur la période)	<ul style="list-style-type: none"> • Catégorie • Ressource/Ressource • Période • Atelier • Catégorie Produit
---------------------------------------	---	---

Légende

1. Plusieurs menus au choix : un bilan général ou des menus plus spécifiques (rendements, coûts, rebus/quarantaine, Disponibilité ou Efficacité)
2. Liste des filtres actionnés (Peuvent être supprimés par la croix)
3. Différents filtres pour affiner les indicateurs
4. Ce bouton permettra d'ouvrir l'onglet correspondant à l'indicateur où sont présentés plusieurs détails (Voir feuille Production – rendements ou feuille Production – Autres menus)
5. Menu déroulant pour affiner les indicateurs et surtout réaliser des comparatifs (Voir feuille Production – Comparatif)
6. Choix entre un mode graphique et un mode tableau
7. Des filtres couleurs peuvent être disponibles :

Vert : taux $\geq 90\%$ (ou $< 3\%$ pour les taux d'attente et taux de temps non utilisable)

Orange : $90 > \text{taux} \geq 70\%$ (ou $3 \leq \text{taux} < 10\%$ pour les taux d'attente et taux de temps non utilisable)

Rouge : taux $< 70\%$ (ou $\geq 10\%$ pour les taux d'attente et taux de temps non utilisable)

A la sélection d'un filtre couleur, tous les taux de la page ne correspondant pas à la couleur devront être grisés

Le code couleur devra être propagé au graphique également (si possible)
8. Les types de coûts pourront être : Matières premières, production, sous-traitance ou total
9. Les types d'activités pourront être : Production, nettoyage, préparation ou total
10. L'ensemble des indicateurs devront avoir un champ d'aide où sera renseigné les formules de calculs des indicateurs



Production - Comparatif

Légende

1. Le comparatif « *sur une période* » correspond à l'évolution de l'indicateur sur cette période suivant un pas donné (mois, trimestre (quarter), année...)
On ne compare ici que l'évolution de l'indicateur entre les différents « pas » de la période.
2. Le comparatif « entre deux périodes » correspond à l'évolution de l'indicateur de la première période renseignée à la deuxième période renseignée. Le « Au » permet d'affiner la comparaison. Par exemple, on veut pouvoir comparer deux années non pas sur leur total mais mois par mois, ou trimestre par trimestre. Dans ce cas, on renseignera cette information dans « Au » (mois, trimestre, année...).
3. Choix entre un mode graphique et un mode tableau
4. Des filtres couleurs peuvent être disponibles :
Vert : taux $\geq 90\%$ (ou $< 3\%$ pour les taux d'attente et taux de temps non utilisable)
Orange : $90 > \text{taux} \geq 70\%$ (ou $3 \leq \text{taux} < 10\%$ pour les taux d'attente et taux de temps non utilisable)
Rouge : taux $< 70\%$ (ou ≥ 10 pour les taux d'attente et taux de temps non utilisable)
A la sélection d'un filtre couleur, tous les taux de la page ne correspondant pas à la couleur devront être grisés
Le code couleur devra être propagé au graphique également (si possible)
5. Pour les comparaisons, reprendre la structure des graphiques et des tableaux présenté dans le BI partie vente

BUSINESS INTELLIGENCE – Production

● Bilan Général ● Rendements ● Coûts ● Réduire/plus détaillé ● Disponibilité ● Taux d'efficacité ● Taux de service

Bilan général

Filtres Mai 2017 (X) Circuit (X)

Périodes Atelier Catégorie Ressource Ressource Famille produit Code couleur 4

Réduire

Sur une période 1 Entre deux périodes 2 5

De 01/03/16
A 31/01/17
Pas Mois

Entre Période Et Période Au Choisir

Nombre d'OTs terminés : 850 Nombre d'OTs ouverts : 150

Taux de rendement Synthétique :
Taux de disponibilité * Taux d'efficacité * Taux de conformité
70,00% 68% (+/- 0,0 %)

Graphique Tableau 3

Périodes	2016	2017		
Janvier	77	+ 2,3 %	75	
Février	80	+ 0,6 %	75	- 3,3 %
Mars	69	+ 0,0 %	70	+ 1,4 %
Avril	70	+ 1,5 %	0	
Mai	65	+ 2,3 %	0	+ 0,0 %
Juin	70	+ 0,0 %	0	
Juillet	85	+ 2,3 %	0	+ 0,0 %
Août	41	+ 2,3 %	0	+ 0,0 %
Septembre	75	+ 2,3 %	0	+ 0,0 %
Octobre	60	+ 2,3 %	0	+ 0,0 %
Novembre	79	+ 1,5 %	0	+ 0,0 %
Décembre	87	+ 1,5 %	0	+ 0,0 %

Plus de détails

Taux de disponibilité :
Temps total de production / Temps d'ouverture
75,00% 68% (+/- 0,0 %)

Temps total de production (heure) : 54698 Temps d'ouverture (heure) : 62541

Graphique Tableau 3

Taux d'efficacité :
Temps total réel de production / Temps total théorique de Production
101,00% 68% (+/- 0,0 %)

Temps total réel de production (heure) : 8452 Temps total théorique de production (heure) : 9854

Graphique Tableau 3

Taux de conformité :
(Production - Débuts et quarantaine) / Production
95,00% 68% (+/- 0,0 %)

Total des débuts et quarantaines sur la période : 4526 Total de la production : 15268

Graphique Tableau 3

Taux d'attente:
Temps total d'attente / Temps d'ouverture
15,00% 20% (+/- 0,0 %)

Temps total d'attente (heure) : 54698 Temps d'ouverture (heure) : 62541

Graphique Tableau 3

Taux de temps non perdus:
Temps total d'arrêt / Temps d'ouverture
10,00% 5% (+/- 0,0 %)

Temps total de correctif (heure) : 8452 Temps d'ouverture (heure) : 9854

Graphique Tableau 3

Taux de service :

Taux de service interne (Date réalisée / Date planifiée)
75,00% 85% (+/- 0,0 %)

Graphique Tableau 3

Taux de service client (Date réalisée / Date demandée)
82,00% 92% (+/- 0,0 %)

Graphique Tableau 3

Coûts : Taux des coûts réels / coûts théorique
102,00% 112% (+/- 0,0 %)

Total coûts réels sur la période : 2 000 505,00€ Total coûts théorique sur la période : 1 985 050,00€

Type de coûts Graphique Tableau 3

Barnabas Csomor 53 2017

Production - Rendements

Légende :

1. Attention, pour ces graphiques :

Dans le cas d'un comparatif « sur une période », la comparaison ne sera pas faite sur les graphiques correspondant. Il sera seulement affiché la valeur totale sur l'ensemble de la période.

Dans le cas d'un comparatif « entre période », la comparaison se fera seulement sur les deux périodes. La granularité (le champs « Au ») ne sera pas pris en compte

2. Si des comparaisons « Entre périodes » sont réalisées, ce graphique ne devra pas être impacté.

3. Choix entre un mode graphique ou tableau Pour les comparaisons, reprendre la structure des graphiques et des tableaux présentés dans le BI partie vente

4. Les croix oranges correspondent aux valeurs du taux sur la période précédente ou sur une des périodes si une comparaison de type « entre deux périodes » a été réalisée.

5. Pour la répartition du taux, celle-ci pourra être faite suivant l'ensemble des répartitions du taux spécifié dans la feuille Indicateurs – Production. Ainsi des graphiques pourront être rajoutés à la page.

Par exemple, pour le taux ci-contre, pourra être rajouté des graphiques pour la répartition par atelier et par catégorie produit

Business Intelligence – Production

(Bilan Général ● Rendements ○ Coûts ○ Rebus/Quarantaines ○ Disponibilité ○ Efficacité ○ Taux de service)

Rendements Synthétiques

Filtres Mai 2017 (X) Circuit (X)

Périodes Atelier Catégorie Ressource Ressource Famille produit Code couleur ■ ■ ■

(Réduire)

Sur une période De 01/03/16
A 31/01/17
Pas Mois

Entre deux périodes Entre Période
Et Période
Au Choisir

Rendements synthétique

• Taux de rendement Synthétique :
Taux de disponibilité * Taux d'efficacité * Taux de conformité
70,00% 68% (période précédente)

● Evolution du taux de rendement synthétique

• Taux générale

Graphique Tableau (3)

Mois 2017	Taux de rendement synthétique (%)
Janvier 2017	~85
Février 2017	~95
Mars 2017	~80
Avril 2017	~75
Mai 2017	~85
Juin 2017	~85
Juillet 2017	~75
Août 2017	~85
Septembre 2017	~85
Octobre 2017	~85
Novembre 2017	~85
Décembre 2017	~85

• Taux détaillé (2)

Mois 2017	Taux de rendement synthétique (%)	Taux de disponibilité (%)	Taux d'efficacité (%)	Taux de qualité (%)
Janvier 2017	~75	~85	~95	~95
Février 2017	~78	~80	~98	~98
Mars 2017	~82	~75	~95	~95
Avril 2017	~75	~70	~95	~95
Mai 2017	~65	~65	~95	~95
Juin 2017	~70	~75	~95	~95
Juillet 2017	~75	~80	~95	~95
Août 2017	~80	~85	~95	~95
Septembre 2017	~85	~80	~95	~95
Octobre 2017	~85	~85	~95	~95
Novembre 2017	~80	~75	~95	~95
Décembre 2017	~75	~70	~95	~95

● Répartition du taux de rendement synthétique (5)

• Catégorie ressources (1) (4)

Catégorie Ressources	Taux de rendement synthétique (%)
Opérateur test	~80
Opérateur montage	~70
Machine sclage	~70
Machine montage	~90
Machine test	~80
Machine CMS	~75

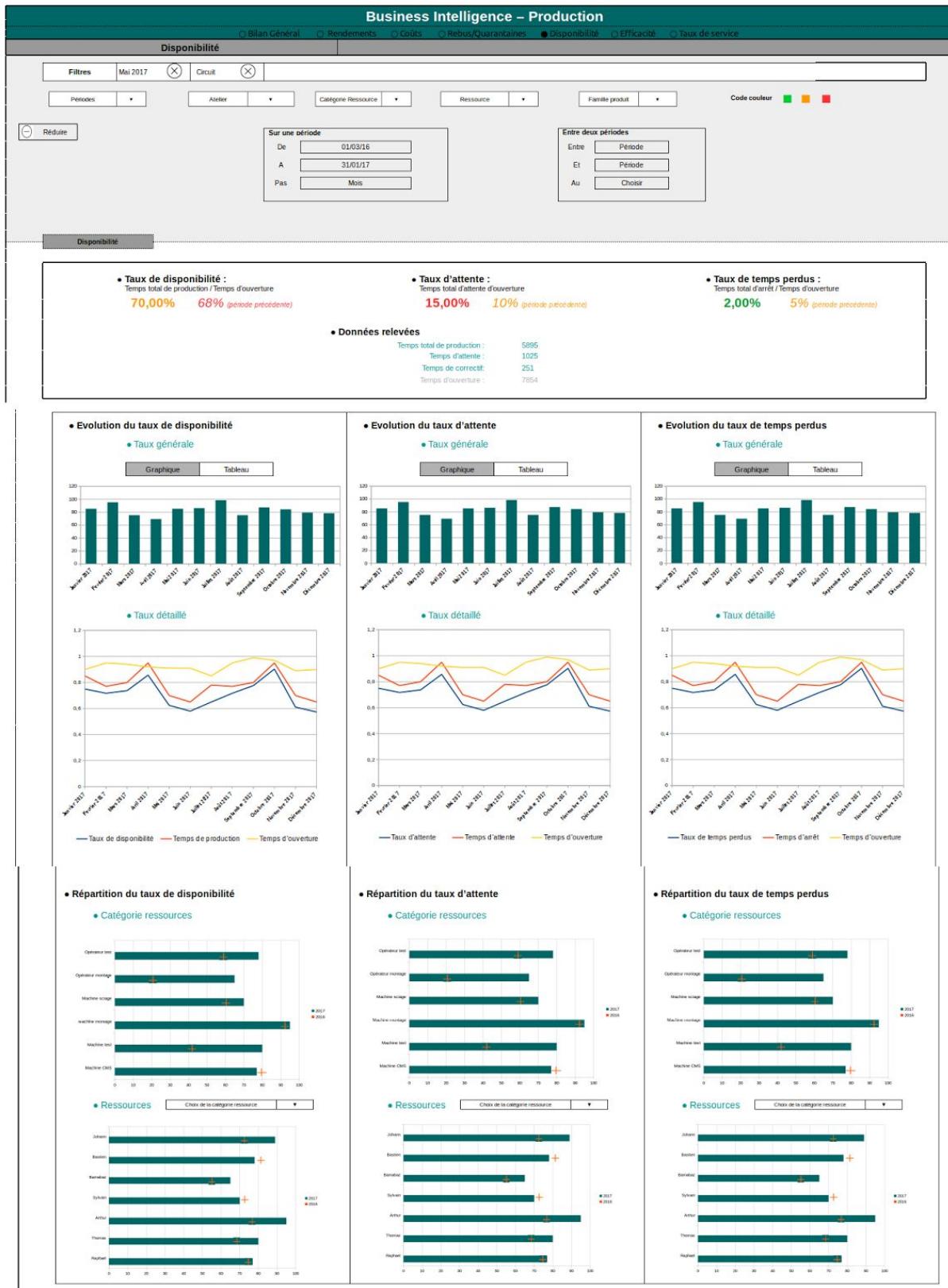
• Ressources (1) (4)

Ressources	Taux de rendement synthétique (%)
Johann	~90
Bastien	~85
Barnabas	~70
Sylvain	~75
Arthur	~95
Thomas	~85
Raphael	~80

Production - Disponibilité

Légende

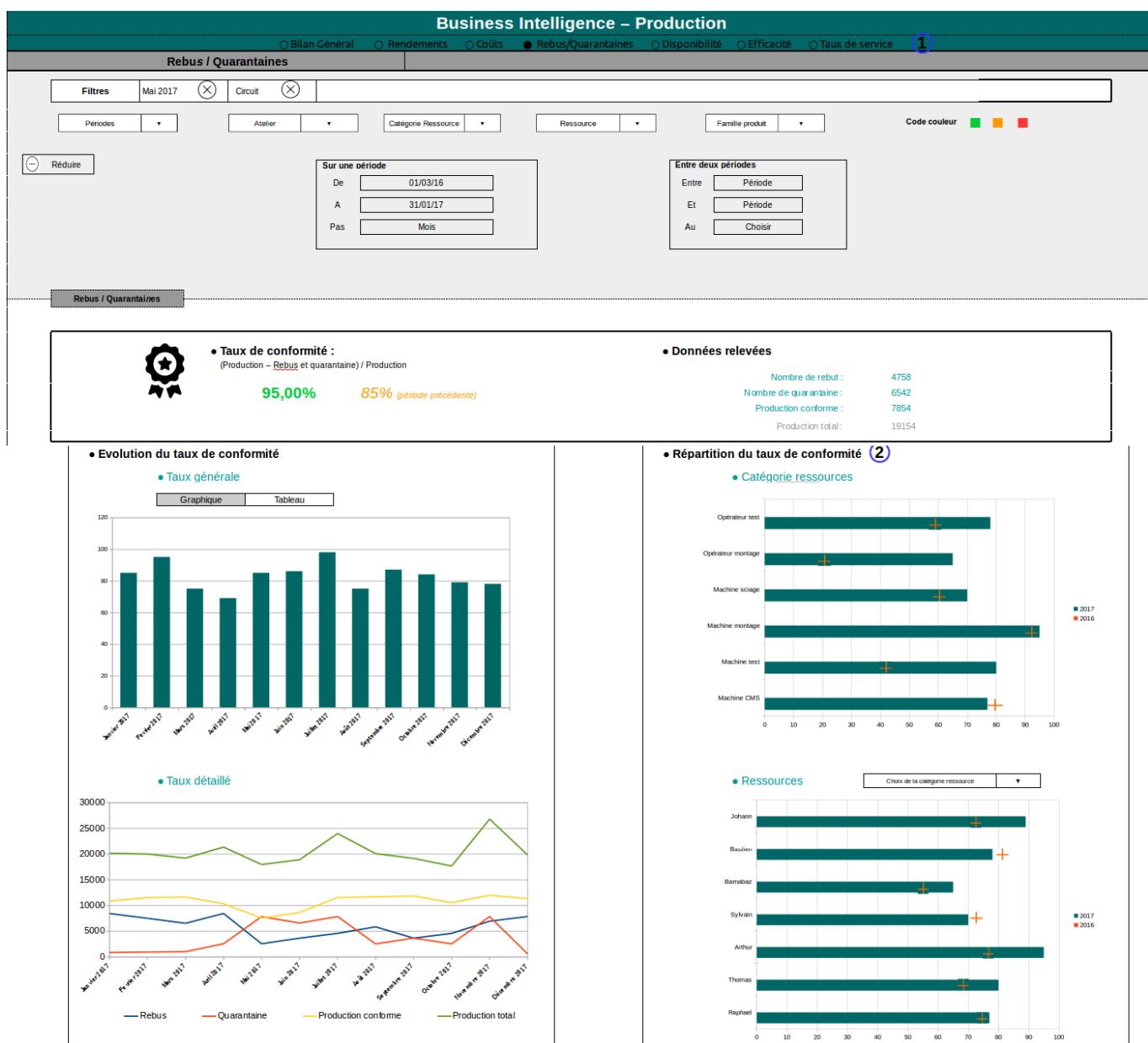
1. Pour la répartition du taux, celle-ci pourra être faite suivant l'ensemble des répartitions du taux spécifié dans la feuille Indicateurs – Production. Ainsi des graphiques pourront être rajoutés à la page.
Par exemple, pour les taux ci-contre, pourront être rajoutés des graphiques pour la répartition par atelier et par catégorie produit



Production -Coûts, Rebus, efficacité et taux de service

Légende :

- Pour chacun des menus Coût, Rebus/Quarantaine, efficacité et taux de service, reprendre la même mise en page que celle ci-contre.
Les conditions sur les graphiques sont indiquées dans la légende de la feuille Production – Rendements
- Pour la répartition du taux, celle-ci pourra être faite suivant l'ensemble des répartitions du taux spécifié dans la feuille Indicateurs – Production. Ainsi des graphiques pourront être rajoutés à la page.



Annexe III

Rendu PDF des tableaux de bord

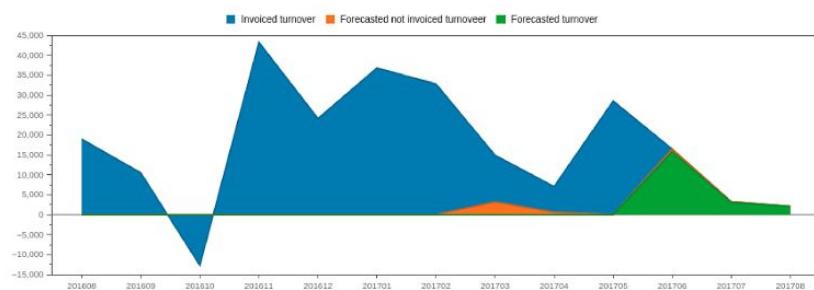
Buisness Intelligence - Sale

Sales performance

Product Category	Product	Analytical Account	Start	End
All categories	All product	All analytical account	08/2016	08/2017
Customer Category	Customer	Geographical area		
All categories	All customer	All country		
Seller				
All seller				

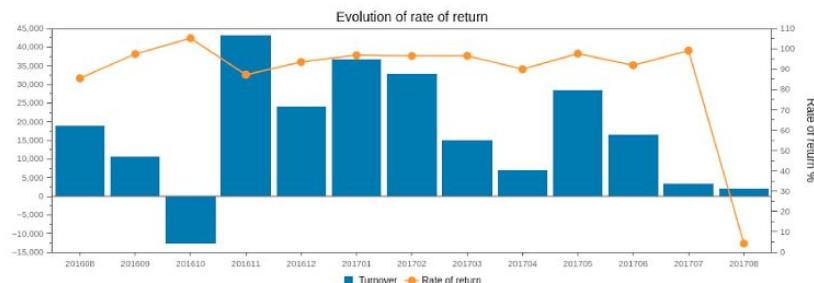
Turnover

Turnover
226013.71 €
 +3870.86 €
(compared to the previous period)



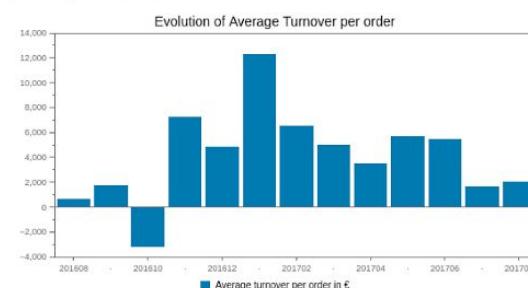
Main Data

Rate of return
87.77 %
 94.38 %
(over the previous period)



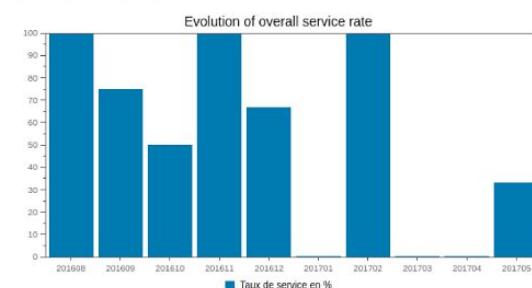
Average Turnover per order
4117.36 €

4726.44 €
(over the previous period)



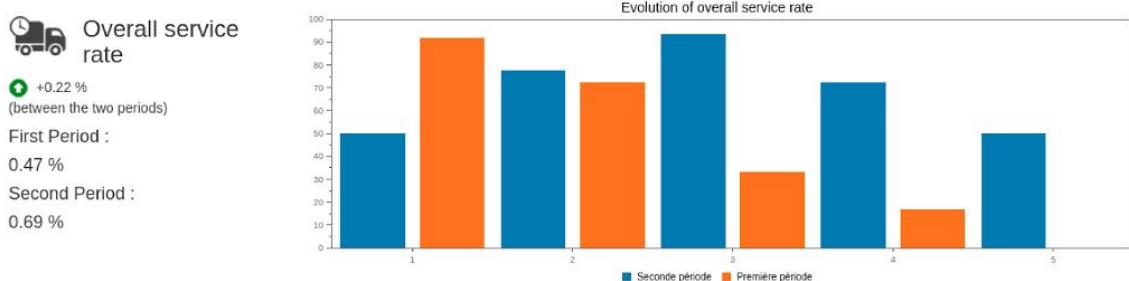
Overall service rate
52.5 %

67.21 %
(over the previous period)



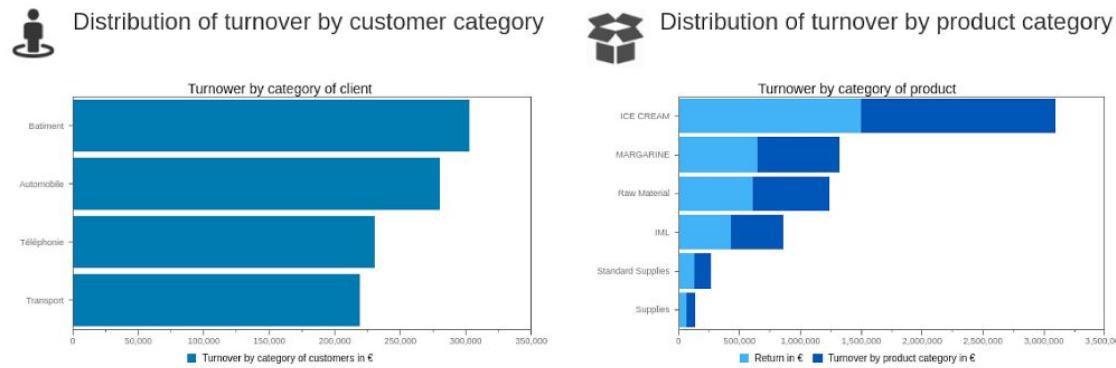
Comparatif du taux de service

Number of periods	
One period	Two period
Step	
Month	
Beginning of the first period	08/2016
End of the first period	08/2017
Beginning of the second period	02/2015
End of the second period	02/2016



Période 1	Période 1 - Taux de rentabilité	Période 2	Période 2 - Taux de rentabilité	Evolution
2016-Q3	0.83 %	2015-Q1		0.6 % -0.23 %
2016-Q4	0.78 %	2015-Q2		0.69 % -0.09 %
2017-Q1	0.43 %	2015-Q3		0.88 % +0.45 %
2017-Q2	0.29 %	2015-Q4		0.78 % +0.49 %
2017-Q3	0 %	2016-Q1		0.5 % +0.5 %

Distribution of turnover and gross profit



Distribution of turnover in the world

Distribution of turnover by country



Other



Outstanding exceeded bill
336620.26 €



Outstanding unpaid bill
984532563.23 €



List of current orders
18777.24 €

Order Number	Date	Product	Customer	Ordered quantity	Invoiced quantity	Delivered quantity	Total
SO1701024	2017-03-05	Tub RW 7.5 CTY CRK SPD SLV	Bill	5000	4000	4000	2086.36 €
SO1701024	2017-02-27	Lid RW 7.5 ICB Yellow	Bill	5000	4000	4000	2086.36 €
SO1701024	2017-03-21	Lid BR 28.5oz RapCheese	Bill	6000	6000	500	2086.36 €
SO1701024	2017-03-08	Lid BR 28.5oz RapCheese	Bill	5000	4000	4000	2086.36 €
SO1701024	2017-03-04	Lid BR 28.5oz RapCheese	Bill	5000	4000	4000	2086.36 €
SO1701024	2017-03-06	Label Tub RW 7.5 ICB Light	Bill	900	0	0	2086.36 €
SO1701024	2017-03-20	Label Tub RW 7.5 ICB Light	Bill	6000	0	6000	2086.36 €
SO1701024	2017-03-02	Label Tub RW 7.5 ICB Light	Bill	5000	4000	4000	2086.36 €
SO1701024	2017-03-10	Label Tub GL Vancar Canada	Bill	5000	4000	4000	2086.36 €

Buisiness Intelligence - Vente

Force de vente

