



HAL
open science

ANALYSE TOPOLOGIQUE DES STRUCTURES DE DONNEES : APPLICATION A LA MIGRATION DES SYSTEMES D'INFORMATION D'ENTREPRISE

François Loison, Piers Barrios, Christophe Danjou, Benoit Eynard

► **To cite this version:**

François Loison, Piers Barrios, Christophe Danjou, Benoit Eynard. ANALYSE TOPOLOGIQUE DES STRUCTURES DE DONNEES : APPLICATION A LA MIGRATION DES SYSTEMES D'INFORMATION D'ENTREPRISE. 13ème CONFERENCE INTERNATIONALE DE MODELISATION, OPTIMISATION ET SIMULATION (MOSIM2020), Nov 2020, AGADIR (virtuel), Maroc. hal-03177804

HAL Id: hal-03177804

<https://hal.science/hal-03177804>

Submitted on 23 Mar 2021

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

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

ANALYSE TOPOLOGIQUE DES STRUCTURES DE DONNEES : APPLICATION A LA MIGRATION DES SYSTEMES D'INFORMATION D'ENTREPRISE

François Loison¹, Piers Barrios^{1,2}

francois.loison@gfi.fr, piers.barrios@utc.fr

¹Gfi Informatique, France

²Université de technologie de Compiègne, France

Christophe Danjou³, Benoit Eynard²

christophe.danjou@polymtl.ca, benoit.eynard@utc.fr

³Polytechnique Montréal, Canada

RESUME : La migration des données de gestion de cycle de vie des produits demande de maîtriser les projets de transformation de données impliquant d'importants volumes de données complexes. Généralement, la complexité est liée à la topologie du graphe de relation entre les objets et les liens.

Lors du cadrage d'un tel projet, il faut être capable de mesurer sa complexité en un temps très court afin d'estimer la charge nécessaire de manière fiable. D'autre part, certains projets de transformation de données présentent un haut degré de complexité. Leur analyse demande une approche holistique et une remise en cause des lotissements de données ou de l'organisation des acteurs du projet.

Ces besoins ont engendré l'initiative Data Systemizer qui a élaboré une méthode outillée dédiée aux projets de transformation de données permettant de faire émerger le meilleur lotissement de données et de le représenter graphiquement dans un formalisme naturel aux décideurs et de fournir des livrables clefs permettant de maîtriser l'exécution du projet. Le « graphe de données inextricable » doit ainsi devenir un ensemble clair et synthétique de lots liés entre eux, celui-ci étant optimisé mathématiquement pour minimiser les dépendances entre lots.

MOTS-CLES : Gestion du cycle de vie du produit, Gestion des connaissances, Industrie 4.0, Systèmes cyber-physiques /IoT (Internet of Things), Système d'information d'entreprise, Modèle de données

1 INTRODUCTION

Un système d'information PLM (SI PLM) est un système d'information dédié à la gestion du cycle de vie produit. Selon (Abramovici et Schulte, 2007), "PLM is an integrated approach including a consistent set of models, methods and IT tools for managing product data, engineering processes, and tools throughout the product lifecycle."

Les processus métier de gestion des données produit, qu'ils soient normatifs ou spécifiques à une entreprise évoluent constamment. Les technologies de l'information et de la communication (TIC) sur lesquelles repose le SI PLM évoluent elles-aussi. Un SI PLM est donc soumis à une double pression métier et technique qui rend nécessaire une migration du système tôt ou tard.

Le cas de la migration est généralement compliqué. Le système source et le système cible ne sont pas forcément bâtis sur la même solution produit ce qui induit un redéveloppement complet du système. Certaines migrations fusionnent plusieurs systèmes source et un système cible, ce qui engendre une complexité accrue. La mécompréhension ou la sous-estimation de la complexité engendre des erreurs d'estimation de la charge du projet et par la suite des délais et surcoûts lors de l'exécution du projet (Friedman, 2019).

Comparée à d'autres systèmes d'information d'entreprise tels que les logiciels de pilotage de la production (MES) ou les progiciels de gestion intégré (PGI – ERP), la migration des données PLM est fondamentale dans le sens où la donnée PLM est ontologiquement constitutive des structures de données techniques (Bricogne *et al.*, 2014) (Eynard *et al.*, 2004).

Les systèmes PLM et se distinguent du cas général par la complexité de leurs liens (par exemple des liens sur entre un objet et un lien) et le nombre de liens qui augmente la complétude des graphes de données.

La mise en œuvre de la migration de données PLM est difficile parce que :

- La stratégie de migration de données est dimensionnante mais est généralement élaborée à partir d'une réalité théorique produites par des sachants qui peut être très éloignée de la réalité des données.
- Un outillage permettant de migrer effectivement les données est nécessaire (Friedman, 2019), l'expérience de nombreux projets de migration de données montre qu'il est en général spécifique.
- Les méthodes projets (En V, Agile ou Scrum) préconisent des approches itératives mais n'expliquent pas comment définir les itérations dans le cas de la migration de données.

Un élément clef de la stratégie de migration est la définition du phasage de la migration en procédant par des lots

séparés (Friedman, 2019). L'identification de ces lots est analogue à une opération de lotissement des données (clustering), tout en minimisant les dépendances entre les lots. L'article définira un critère de mesure de la qualité du lotissement afin qu'il puisse engendrer une migration de données efficace. Ce critère permet de comparer des stratégies de migration de données.

Pour répondre à ces difficultés, notre article s'intéressera à une méthode outillée permettant de définir, simuler et comparer plusieurs stratégies de migration de données tout en fournissant des livrables clefs permettant de maîtriser l'exécution du projet. Il sera structuré de la manière suivante : nous nous focaliserons sur l'état de l'art de l'interopérabilité et la migration des données PLM. La proposition de méthode outillée soulignera les difficultés rencontrées et les choix faits pour atteindre une solution utilisable. Cette proposition sera illustrée par des cas d'étude industriels. Pour finir, une discussion, une conclusion et une vue d'ensemble des travaux futurs complètera cet article.

2 DE L'INTEROPERABILITE A LA MIGRATION DES DONNEES

2.1 Interopérabilité

Le cadre européen d'interopérabilité (EIF, 2017) définit quatre dimensions : légale, technique, sémantique et organisationnelle. L'interopérabilité sémantique est la clef ; « Semantic interoperability is the key one here; it is the ability to automatically interpret the information exchanged meaningfully and accurately in order to produce useful results as defined by the end users of the systems under consideration » (Vernadat, 2010).

(ISO 14258, 1998) explique que l'interopérabilité sémantique peut être atteinte de trois manières :

- Intégrée : « Avec les modèles intégrés, il existe une forme de modèle standard » La principale limite est ici la standardisation des modèles.
- Unifiée : un méta-modèle existe et représente l'ensemble des modèles existants. Par conséquent, il est possible d'établir une correspondance entre les divers modèles.
- Fédérée : les modèles sont hétérogènes et reposent sur l'établissement de correspondances dynamiques.

Pour approfondir sur le sujet de l'échange de données et de l'interopérabilité, le lecteur peut consulter les articles suivants pour un état de l'art plus détaillé : (Rachuri *et al.*, 2008), (Fortineau *et al.*, 2013), (Sriti *et al.*, 2015), (Leal *et al.*, 2019) et (Danjou *et al.*, 2017).

Enfin, l'intégration est la façon la plus sollicitée d'aboutir à l'interopérabilité au sein du PLM dans l'industrie, or pour qu'une intégration se réalise, il faut nécessairement passer par une migration des données.

Concernant les normes existantes qui pourraient être considérées sont principalement l'ISO 10303 et notamment le protocole d'application AP239 aussi connu sous

le nom de PLCS et la dernière version de l'AP242 visant l'interfaçage des solutions PLM et CFAO. D'autres travaux intéressants sont ceux relatifs à la norme ISO 13584 aussi appelée PLib mais qui malheureusement datent désormais un peu et n'ont pas connu une exploitation effective.

2.2 Migration des données PLM

Les recherches au travers de Scopus, Web of Science et de Google Scholar ont mis en exergue les éléments suivants.

Il existe des documents relatifs à la migration de données dans le domaine général (non spécifique PLM).

Les travaux retenus sont :

- (Friedman, 2019) propose des éléments méthodologiques pour réussir une migration de données. L'article propose une approche minimaliste limitant le nombre de sources de données et de données à migrer, insiste sur l'évaluation d'un phasage de migration et l'utilisation d'outils efficaces. Il est recommandé de se focaliser le plus tôt possible sur la qualité des données.
- (Wu *et al.*, 1997) propose différentes méthodes intéressantes de migration de données permettant de structurer un projet de migration de données. La méthode *Butterfly* propose de migrer graduellement les données, ce qui correspond à une approche itérative, mais n'explique pas comment déterminer des lots de données applicables.
- (Kuncheria, 2007) met en évidence les points clefs d'une migration de données sans toutefois les aborder dans ces termes : « Migration of data from legacy enterprise data-systems is one of the important and challenging tasks for most PLM implementations. [...] Moreover, the dependency of production activities on data, that is complete and accurate, stresses the importance for error free data migration. Translating data stored in formats as required or limited by legacy systems to conform to the PLM system specifications only adds to the complexity. » Cependant, au travers de l'article, le processus de migration de données réel n'est ni explicité, ni détaillé.

Aucun de ces travaux ne nous permet concrètement d'élaborer et de vérifier une stratégie de migration de données, si possible reposant sur une approche itérative et en procédant par phases, par lots de données séparés (Friedman, 2019).

3 DATA SYSTEMIZER : UNE METHODOLOGIE DE MIGRATION DES DONNEES

La problématique consiste à produire plusieurs stratégies de lotissement de données (clustering) simulées à partir des données sources et de les comparer afin de produire

des éléments décisifs permettant de choisir la meilleure stratégie de migration de données.

Pour répondre à cette problématique, nous proposons une méthode outillée standard et réutilisable concrétisée par un atelier de génie logiciel nommé Data Systemizer qui repose sur deux principes fondateurs :

- Découper un problème complexe en plusieurs problèmes plus simples tout en optimisant le découplage des sous-systèmes. En appliquant ce principe à la problématique étudiée, il s'agit de découper les données en lots de données les plus indépendants possibles.
- Simuler la stratégie et le processus de migration de données à l'aide d'un outil disposant d'un méta-modèle unifié mis en correspondance avec les modèles de données source (multiples ou pas) et cible.

Le Data Systemizer nécessite un modèle de données spécifiquement conçu pour s'adapter rapidement aux multiples grammaires et modèles de données source existant et être concis pour faciliter le paramétrage des analyses et être performant. Nous présenterons un premier modèle permettant d'effectuer une seule passe de lotissement (regroupement) produisant des lots d'objets et de liens ainsi que des dépendances de lots. Nous verrons par la suite comment généraliser ce modèle pour que le même modèle puisse supporter N passes de lotissement. Nous traiterons du modèle de données en premier lieu puis aborderons le lotissement des données.

Note : pour tous les graphes de lotissement présentés, Les rectangles sont des lots de données, leur label contient les nombres d'objets et de liens inclus. Les flèches sont des dépendances de lots, leur label contient le nombre de liens. La couleur des lots ou dépendances de lots est significative et est expliquée pour chaque illustration.

3.1 Modèle de données concis et généralisé

Les SI PLM ayant tendance à embarquer le plus d'information possible, le nombre d'attributs par entité (objet ou lien) est élevé.

L'utilisation d'un tel modèle aurait l'inconvénient suivant : le nombre de types et d'attributs trop élevé demande une activité de paramétrage du modèle du Data Systemizer trop importante. Les études étant réalisées en 2 à 6 semaines, cette surcharge n'est pas acceptable donc le modèle de données du Data Systemizer doit être simplifié pour ne garder que l'essentiel nécessaire aux analyses, cette approche étant par ailleurs bénéfique à la performance.

Afin de pouvoir réaliser des choix sur la topologie des données, seul un sous-ensemble des entités et de leurs attributs est requis. Le modèle du Data Systemizer ne doit contenir que les entités et les attributs nécessaires au lotissement des données. Le modèle résultat est qualifié de concis.

Les SI PLM ont également tendance à ramifier les arborescences de types. Cette tendance multiplie le nombre de types du système. Pour généraliser le modèle, nous regroupons tous les types enfants en une seule entité parente afin de faciliter la définition des règles d'analyse. Le modèle concis et généralisé respecte l'approche « intégrée » de l'interopérabilité sémantique. En pratique, cette approche a toujours fonctionné, quelle que soit la diversité des modèles de données client ou la diversité des modèles de solution éditeur.

3.2 Modèle logique homogène

Le modèle logique du Data Systemizer permet d'affecter des objets et liens entre objets à des lots. Voici un aperçu d'une première ébauche de modélisation :

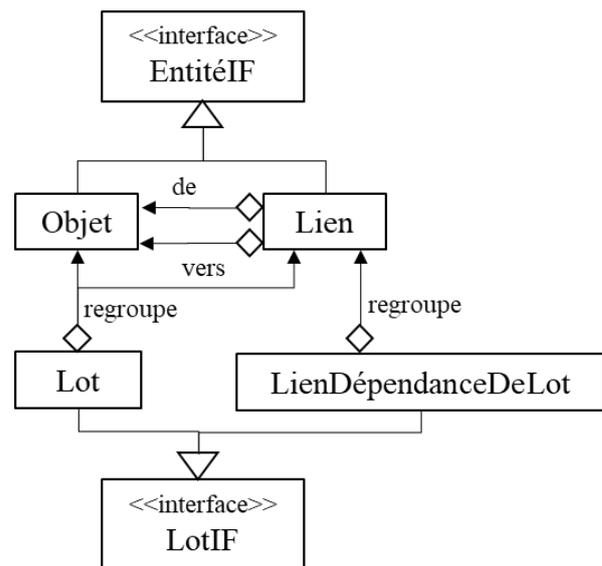
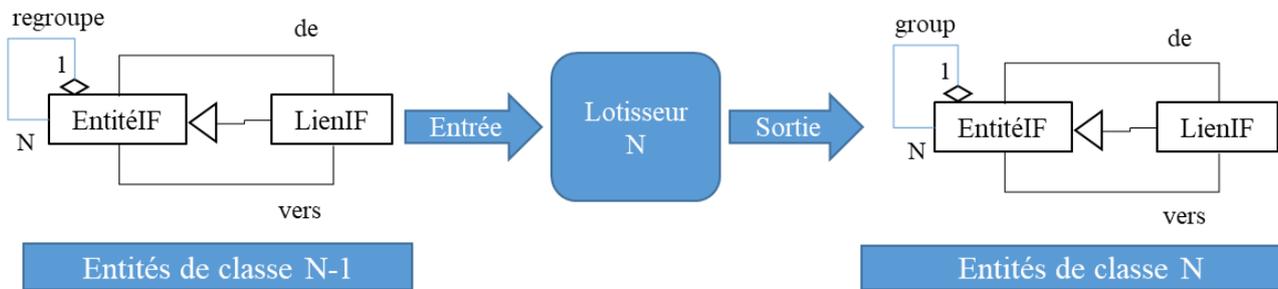


Figure 1 - Ebauche de modélisation

Cette modélisation définit des objets et des liens, qui implémentent l'interface *EntitéIF*. Un lien binaire pointe vers deux objets. Pour produire des lots de données, le Data Systemizer applique des algorithmes de lotissement sur les entités, ils créent des lots et assignent les entités aux lots. Dans le cas de lots interdépendants, les liens entre deux lots sont affectés à un type de lot particulier nommé *LienDépendanceDeLot* qui ne peut contenir que des liens.

Ce modèle ne fonctionne que pour une passe de lotissement car les lots et les dépendances de lots ne sont pas des entités. Nous souhaitons que le modèle puisse effectuer plusieurs passes de lotissement. On constate qu'il existe un isomorphisme: *Objet* est isomorphe à *Lot*, *Lien* est isomorphe à *LienDépendanceDeLot*. Pour généraliser le modèle, on définit la classe d'une entité comme le numéro de la passe du lotissement et on déclare que les lots et les dépendances de lot sont aussi des entités.



Le modèle dénommé « Entité/Lien » est représenté sur la **Figure 2**.

Le tableau suivant précise les entités et leurs classes correspondantes :

Élément	Classe d'entité
Objet ou lien	Entité ^[0]
Lot ou Dépendance de Lot contenant des Entités ^[N-1]	Entité ^[N]

Figure 2 - Modèle homogène

Le modèle Entité/Lien est homogène : il est identique par isomorphisme après une passe de lotissement, ce qui permet d'enchaîner une nouvelle passe de lotissement sans rupture de modèle. Cette caractéristique simplifie fortement la mise en œuvre du moteur de lotissement.

3.3 Lotissement

Lotissement est l'équivalent en sémantique métier PLM de « regroupement » ou « clustering ». L'objectif du lotissement est en premier lieu de répondre aux besoins du client, par exemple de lotir les données par programmes métier (famille de produits) ou par exemple par site physique. Dans un deuxième temps, il faut vérifier que le lotissement est phasable. Par exemple, un lotissement en sites physique avec de nombreuses dépendances bidirectionnelles entre les sites physiques n'est pas phasable, un site physique n'est pas migrable individuellement car les données résultantes ne seront pas complètes puisque les liens vers l'autre site n'auront pu être migrés.

Les critères expliqués ci-après permet de mesurer et comparer l'efficacité de lotissements de données :

- Faible couplage des lots de données pour pouvoir paralléliser le traitement des lots de données.
- Lots de données les plus significatifs fonctionnellement : les données et leurs dépendances doivent contenir un ou plusieurs produits métiers complets pour permettre la validation métier.
- Les lots doivent être décomposés en un arbre de lots faiblement couplés de manière à pouvoir appliquer récursivement ce processus au niveau des sous-lots pour produire efficacement des lots de données de test pour une volumétrie et un taux d'erreurs donnés.

Le Data Systemizer est conçu pour paramétrer rapidement un ou plusieurs lotissements et de mesurer leur efficacité selon les critères expliqués précédemment. Il propose des « lotisseurs » qui sont des algorithmes qui

prennent des entités en entrée et produisent des lots et dépendances de lots en sortie. Le lotissement est constitué par l'assemblage d'une séquence de lotisseurs.

Il y a deux catégories de lotisseurs :

- Lotisseur métier : ils utilisent des paramètres métier fournis explicitement par le métier et disponibles dans les données. Par exemple : lotir les données par le nom d'un programme (ensemble de produits métier), lotir les données par site de fabrication, lotir les données par intervalle de date de dernière modification.
- Lotisseur mathématique : il analyse la topologie du graphe et ne requiert pas de paramètre métier. Par exemple : le lotisseur Metis effectue une partition du graphe fourni en sous-graphes tout en réduisant les dépendances des sous-graphes ; le lotisseur Bill Of Material (BOM) groupe les entités jusqu'à ce qu'une entité soit réutilisée.

Metis (Abou-Rjeili et Karypis, 2006) est un outil de partitionnement de graphe prenant en entrée un graphe et le nombre souhaité de partitions N. Il produit N sous-graphes optimisés pour réduire le nombre d'arêtes joignant deux partitions. Metis est utilisé par la stratégie de lotissement « meilleur que Big Bang » ayant pour but de décomposer les données en lots les plus indépendants possible afin de réduire la complexité de la migration.

Les lotisseurs peuvent être assemblés sans contrainte technique, ce qui produit une forte combinatoire de stratégies de lotissement possibles. Il est ainsi possible de produire plusieurs graphes de lotissements afin de choisir celui qui est le plus en adéquation avec les besoins clients et qui est le plus efficace.

3.4 Connexion à l'ETL

Une fois l'analyse du lotissement des données effectuée et validée, il faut migrer effectivement les données vers le système cible en respectant le phasage défini par la stratégie de migration de données.

L'état de l'art actuel consiste à effectuer un traitement par l'ETL de lots de taille importante, par exemple de 1M de données et de liens. Une défaillance sur le traitement une entité va entraîner des défaillances sur les liens dépendants de cette entité. Le phénomène se poursuit produisant des erreurs en cascade ayant des impacts négatifs sur le traitement du lot : il n'est pas testable fonctionnellement et un grand nombre d'erreurs est reporté (600k constaté pour 1M de données en entrées). Il faut

alors identifier les causes racines de erreurs ce qui demande un effort d'analyse important.

Le Data Systemizer dispose d'une interface permettant de piloter l'outil ETL réalisant effectivement la migration de données. Cette interface fournit à l'ETL des fichiers par lots contenant les données avec une grammaire identique à celle des fichiers source. Elle lance une opération ETL sur un lot données et analyse les résultats de l'opération pour décider si les lots suivants doivent être traités ou s'il faut interrompre l'opération, afin d'éviter les erreurs en cascade.

4 APPLICATION

4.1 Simuler la fusion de deux systèmes sources dans un PLM cible

Un client souhaite fusionner des données *as-designed* et *as-built* provenant de deux systèmes distincts dans un PLM cible. Pris séparément, les systèmes existants sont intègres. Comme il n'existe pas de vérification systématique de la cohérence des données *as-designed* et *as-built*, le client suppose qu'il y aura des données irréciliables ou erronées. Cependant leur volumétrie et leur identification n'est pas qualifiée ce qui empêche de dimensionner les chantiers de nettoyage de données de systèmes source et dans une certaine mesure des activités de migration.

L'état de l'art actuel consiste à démarrer le projet de migration et de mesurer la qualité de réconciliation des données après la fabrication et l'exécution de plusieurs itérations de l'outil ETL cible. Le nettoyage des données démarre tardivement et se retrouve dans le chemin critique du projet de migration.

Le Data Systemizer est utilisé pour simuler la réconciliation des données en phase d'étude de la migration de données. Les résultats permettent de lancer le chantier de nettoyage des données plus tôt.

Après paramétrage des règles de réconciliation, le Data Systemizer simule la réconciliation des données et donne un état complet des erreurs de réconciliation rencontrées. Le Data Systemizer est paramétré pour réconcilier les données, rechercher les erreurs et regrouper les données par îlots le plus indépendant possibles.

La **Figure 3** montre le graphe du contenu d'un îlot de données indépendant mis en évidence par l'outil.

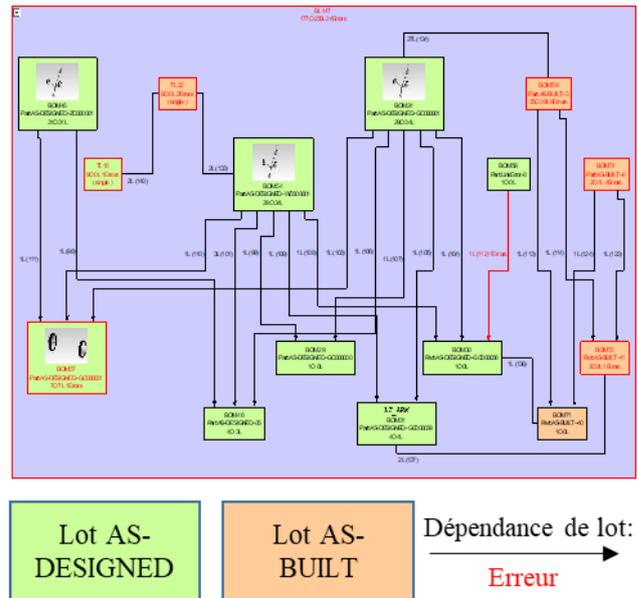


Figure 3 : Contenu d'un îlot de données

Des lots issus de la réconciliation *as-designed* / *as-built* ont été produit pour le lot métier concerné (lot englobant du graphe). Une dé-décomposition plus fine a été produite. Les 3 lots vert clair avec une image de charriot de golf sont des lots contenant le haut de l'arborescence d'un produit métier. L'un d'entre peut être sélectionné pour exporter un jeu de test fonctionnement significatif, qui permettra d'élaborer et de tester l'ETL avec un temps d'itération court.

En synthèse :

- Le Data Systemizer a simulé la réconciliation et le chargement des données dans le PLM cible. Il a identifié les erreurs de réconciliation et d'intégrité référentielle (liens orphelins) et a permis de dimensionner et alimenter le chantier de nettoyage des données.
- La stratégie de migration des données a été confortée par la mise en évidence d'îlots de données indépendants et extractibles.
- Certains îlots de données sont devenus des jeux de données de test qui ont permis d'initier le développement itératif de la chaîne de production ETL¹.

4.2 Mesurer la qualité topologique d'un déploiement par « Data Packages » séquentiels

Le client souhaite migrer des données d'un système original de gestion des exigences vers un système PLM. Les données d'une volumétrie de 5M d'entités sont organisées par Data Packages (DP) séquentiels qui sont chargés par ordre chronologique.

En appliquant les pratiques de l'état de l'art actuel, le chargement prend quatre jours sur le système cible, le

¹ Extract/Transform/Load (ETL) : atelier logiciel spécialisé dans l'exécution de migration de données

mode itératif est coûteux. Les erreurs sont très nombreuses et demandent une analyse *ad hoc* jetable pour identifier les causes racines.

Le client souhaite diminuer le coût et le délai d'une itération de chargement, c'est l'objectif se lequel se focalise le Data Systemizer.

Les principales erreurs proviennent d'erreurs de dépendances chronologiques de chargement des data packages illustrées dans la **Figure 4**.

La **Figure 4** indique un cas correct et un cas incorrect de précedence chronologique des Data Packages. Dans le cas correct toutes les dépendances de DPs suivent la chronologie de chargement des DPs. Dans le cas incorrect, le DP1 utilise des données du DP2 (flèche de dépendance rouge). Le DP1 ne peut pas être chargé puisqu'il qu'il nécessite des données du DP2 qui n'est pas encore chargé.

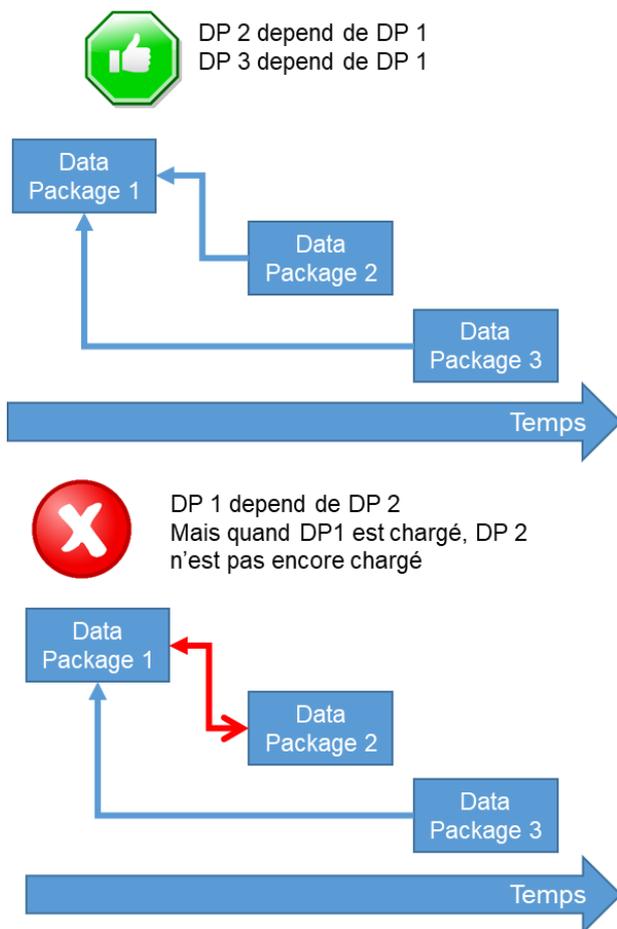


Figure 4 - Illustration des règles de chargement des DP

Le Data Systemizer dispose d'un mode d'analyse chronologique qui est activé. La simulation de la première livraison du client prend vingt minutes et aboutit au graphe suivant :

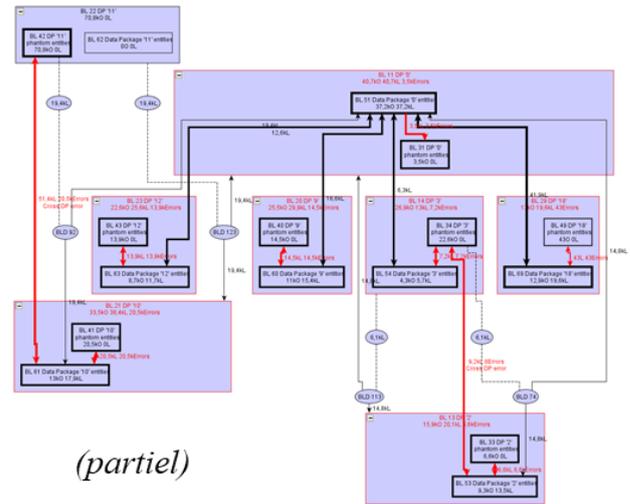


Figure 5 - Première Analyse

La **Figure 5** présente la première analyse effectuée. La couleur rouge indique que des règles de cohérence n'ont pas été respectées. La topologie du graphe de Data Packages est confuse. Le client améliore son processus d'extraction des données à l'aide des rapports fournis par le Data Systemizer.

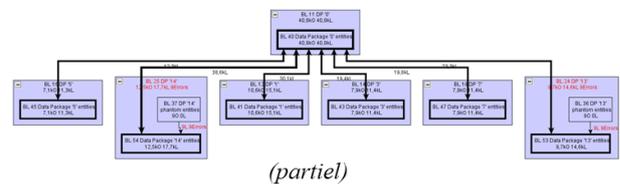


Figure 6 - Quatrième Analyse

La **Figure 6** présente le quatrième essai qui est concluant : La topologie est lisible, les erreurs restantes sont internes aux Data Packages. Un Go pour chargement à blanc est donné, il confirme les résultats de l'analyse.

Ainsi, le Data Systemizer a simulé un chargement par Data Packages chronologique. Le rapport de vélocité est flagrant : 20 minutes contre 4 jours. Les résultats de simulation sont globalement conformes aux tests réels ceci a permis de diminuer la charge des itérations.

5 CONCLUSION ET PERSPECTIVES

Cet article a proposé une méthode outillée permettant par une approche holistique et centrée métier d'analyser et réduire la complexité de la migration de données, tout en fournissant des critères pour mesurer l'efficacité d'une stratégie de migration de données. Cette méthode outillée dispose d'un méta-modèle « unifié, concis, généralisé et homogène » lui permettant de supporter la majorité des modèles PLM client. A l'autre bout de la chaîne, une connexion avec les outils ETL permet de piloter la chaîne ETL en appliquant une stratégie déterminée tout en contrôlant le phénomène d'erreurs en cascade.

Les travaux futurs consisteront à améliorer certains lotisseurs mathématiques (adhérence), à améliorer l'interface

graphique et à mesurer l'efficacité de cette méthode pour faciliter son déploiement. Qui plus est, cette méthode outillée peut s'appliquer à d'autres catégories de systèmes d'information d'entreprise : ERP, MES, SLM².

REFERENCES

EIF, 2017 Communication from the commission to the european parliament, the council, the european economic and social committee and the committee of the regions, European Interoperability Framework – Implementation Strategy, COM/2017/0134 final, <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM:2017:134:FIN> (accessed 04 June 2019).

Abou-Rjeili, A. and G. Karypis, 2006. Multilevel algorithms for partitioning power-law graphs. In: Proceedings 20th IEEE International Parallel & Distributed Processing Symposium. Rhodes Island, Greece, p. 10.

Abramovici, M. (2007) Future trends in product lifecycle management (PLM), Proc. of the 17th CIRP Design Conference, Berlin, Germany, p. 665-674.

Bricogne M., J. Le Duigou, and B. Eynard, 2014. Des SGDT au Product Lifecycle Management, Documentaliste-Sciences de l'Information, 51(1), p. 34-37.

Eynard B., T. Gallet, P. Nowak, and L. Roucoules, 2004. UML based specifications of PDM product structure and workflow, Computers in Industry, 55, p. 301–316.

Danjou, Ch., Le Duigou, J., Eynard, B., 2017, Closed-Loop Manufacturing process based on STEP-NC, International Journal on Interactive Design and Manufacturing, 11(2), p. 233-245.

Fortineau, V., T. Paviot and S. Lamouri, 2013. Improving the interoperability of industrial information systems with description logic-based models-The state of the art. Computers in Industry. 64, p. 363–375.

Friedman T, 2019. Make Data Migration Boring: 10 Steps to Ensure On-Time, High-Quality Delivery. G00448218.

Leal G., W. Guédria and H. Panetto, 2019. Interoperability assessment: A systematic literature review, Computers in Industry, 106111–132.

Rachuri, S., E. Subrahmanian, A. Bouras, S.J. Fenves, S. Fofou and R.D. Sriram, 2008. Information sharing and exchange in the context of product lifecycle

management : Role of standards. Computer-Aided Design. 40, p. 789–800.

Renji Kuncheria, J. 2007. PLM implementation at MAN Diesel A/S: a case study. In: Garetti, M., Terzi, S., Ball, P., Han, S. (eds.) Product Lifecycle Management. Assessing the industrial relevance. pp.199-204 Inderscience, Geneva.

Sriti, M.F., I. Assouroko, G. Ducellier, P. Boutinaud and B. Eynard, 2015. Ontology-based approach for product information exchange. International Journal of Product Lifecycle Management. 8, 1.

Vernadat, F., 2010. Technical, semantic and organizational issues of enterprise interoperability and networking, Annual Reviews in Control, 34, 139–144.

Wu, B., Lawless, D., Bisbal, J., Grimson, J., Wade, V., O'Sullivan, and D., Richardson, 1997. Legacy System Migration: A Legacy Data Migration Engine, Proceedings of the 17th International Database Conference (DATASEM '97), Brno, Czech Republic, October 12, 14, 1997. pp 129-138, Ed. Czechoslovak Computer Experts.

Zhong, H., Yan, G., Lei, Y. 2012. Evolution Supporting Class-Cluster Data Model for PLM. In: Jin, D. and Lin, S. (eds.) Advances in Electronic Commerce, Web Application and Communication. pp. 191–196. Springer, Heidelberg.

² Service Lifecycle Management (SLM)