

Discngine

Utilisation des données au format XML dans GINKGO

Assistance à maîtrise d'ouvrage

Sébastien Conilleau
10/04/2019

Sommaire

1. Description succincte de GINKGO	2
2. Rapport d'étonnement	3
"Pureté" du système	3
Longueur du processus d'actualisation	3
Système laissé à l'état de "work in progress"	3
Data science "old school"	3
3. Les différences entre Tagged file et XML files	4
Structure générale	4
Éléments individuels	5
Cas des corrections	11
4. Evaluation du Scénario « conservateur »	12
Principes du scénario	12
Impact du format XML sur le chargement	12
Impact des différences entre tagged files et XML files sur le chargement	12
Etat des lieux de l'existant (projet 2017)	16
Conclusions sur le scénario conservateur	22
5. Scénario évolutif	22
Description générale	22
Chargement	23
Mise en qualité	24
Création d'un schéma orienté métier	29
Conclusion sur le scénario évolutif	33
6. Conclusion générale	33

Cette étude a pour but d'analyser différents scénarii d'intégration des données fournies par Clarivate Analytics au format XML dans le système GINKGO.

Après une description succincte du système actuel et la présentation d'un rapport d'étonnement elle s'attachera à l'analyse des différences entre tagged files et XML files, puis aux impacts et solutions potentielles de ces différences.

Ensuite elle analysera deux scénarii d'intégration :

- Un scénario « conservateur », qui s'attachera à modifier le système le moins possible ; dans cette partie un état des lieux du projet pilote mené dans ce sens en 2017 sera effectué.
- Un scénario d'évolution du système ; les objectifs d'un tel scénario sont doubles : assurer la continuité de la production des indicateurs publiés en utilisant les données de Web of Science au format XML et diminuer le coût d'actualisation du système, en simplifiant ce qui peut l'être.

1. DESCRIPTION SUCCINCTE DE GINKGO

GINKGO est un système d'informatique décisionnelle. Il a été conçu afin de permettre le chargement, l'analyse et l'exploration de données de différentes sources, notamment des bases de publications (par exemple Web of Science, Scopus) ou des bases de brevets (PATSTAT, Clarivate Derwent).

Il est constitué très majoritairement de schémas de base de données. Le contenu de ces schémas est mis à jour par des jobs Talend (du chargement du DSA jusqu'au chargement du datawarehouse) ou des scripts SQL (en aval du datawarehouse).

L'analyse de GINKGO montre que le système présente cinq grandes fonctionnalités, comme illustré en Figure 1.

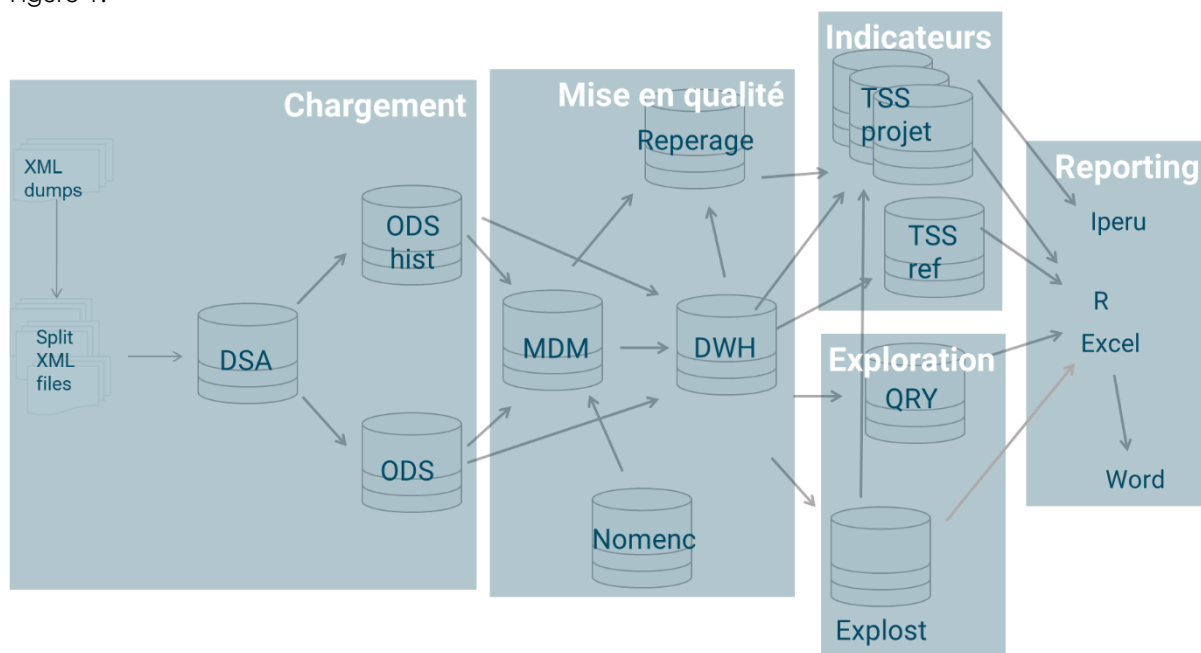


Figure 1 - les cinq unités fonctionnelles de GINKGO

Ces cinq unités sont :

1. Le chargement – l'objectif est de charger en base de données les données brutes initialement livrées sous la forme de fichiers
2. La mise en qualité – cette phase conduit principalement à la mise à jour du datawarehouse (DWH). Les données y sont pour la plupart normalisées. Elles ont fait l'objet de plusieurs mises en qualité comme la réaffectation multidisciplinaire, l'unification des titres¹¹ et l'enrichissement géographique (attribution de codes NUTS et d'identifiants ville-pays). Un repérage institutionnel est aussi effectué mais le résultat est stocké dans un autre schéma Oracle.
3. Exploration – les activités d'exploration sont variées ; on pourra citer la délimitation de périmètres en vue de produire des indicateurs, l'exploration de nouvelles méthodes analytiques, la production de rapports à façon. L'accès aux données à des fins exploratoires se fait majoritairement via des requêtes SQL. La délimitation des périmètres d'études est aussi faite via Explost.

4. Production d'indicateurs – il s'agit d'une activité historique et récurrente de l'OST du Hcéres ; les indicateurs sont stockés dans des tableaux statistiques systématiques (TSS) calculés à partir du datawarehouse. Un jeu de TSS de référence, correspondant au périmètre mondial, est systématiquement calculé. Il permet notamment de disposer rapidement de données de références pour pouvoir comparer des périmètres spécifiques à des périmètres généraux. Un jeu de TSS « locaux » peut être ensuite calculé sur des périmètres faisant l'objet d'une étude.
5. Reporting – toutes les productions du Hcéres, qu'il s'agisse d'études d'impact, d'analyses internes, de développements de nouvelles méthodes ou autre, font l'objet de publications dans des formats ad-hoc. Certaines publications sont largement automatisées (IPERU ou LOLF par exemple), cependant la plupart sont trop spécifiques pour faire l'objet d'une automatisation. Les outils de reporting sont donc laissés à l'appréciation de la personne en charge. Parmi ces outils on trouve très majoritairement R et la suite Microsoft Office.

2. RAPPORT D'ETONNEMENT

“Pureté” du système

La description de haut niveau de GINKGO montre un système de type « informatique décisionnelle » imaginé dans les règles de l'art (DSA – ODS – Datawarehouse – Datamarts). Les systèmes d'informatique décisionnelle ont été imaginés dans un contexte bien précis : celui de l'exploitation de données hétérogènes multi-sources. Par ailleurs ce type d'architecture a un coût d'implémentation et de maintenance assez élevés, justifié par l'enjeu de prise de décisions rationalisées par les données disponibles.

Dans le cas de GINKGO une seule source de données est en fait utilisée, les données sont relativement peu complexes, et l'enjeu décisionnel est minime. Il en résulte un système largement surdimensionné pour l'usage et ce qui a pour conséquence des coûts de mise en place et de maintenance nettement supérieurs au besoin réel.

Longueur du processus d'actualisation

Le processus d'actualisation, de mise en qualité et de recette technico-fonctionnelle est ressenti comme étant très long et coûteux pour les équipes, et ce sans atteindre un résultat satisfaisant quant à la qualité des données finales. Ce ressenti est confirmé par la charge estimée à près de 300 jours homme en 2018 (au 10/14 l'estimation est de 145 pour l'OST, 133 pour le DSI).

Cela est dû notamment à un certain nombre de processus de validation et de corrections manuels (fichiers Excel, exécution de scripts SQL « à la main »). Par ailleurs certaines erreurs de chargement ont pu se reproduire pendant plusieurs actualisations (cas de St Etienne vs St Etienne du Rouvray), ce qui laisse penser qu'il est difficile de faire évoluer le système, notamment au niveau de la mise en qualité de données.

Système laissé à l'état de “work in progress”

De manière assez paradoxale par rapport à la pureté du système évoquée dans le premier point le système souffre de lacunes techniques importantes. On peut mentionner :

- L'absence récurrente de clés étrangères, presque totale d'indexes et parfois même de clés primaires dans les tables
- Une absence de convergence des utilisateurs vers QlikView alors que c'était l'outil qui avait été choisi pour les tâches d'exploration
- L'absence d'outil de navigation dans les TSS, qui sont conceptuellement des cubes OLAP, et qui à ce titre n'ont pas vocation à être requêtés en direct via requêtes SQL mais plutôt à être exposés dans un outil de navigation, de production de rapport ou d'analyse
- L'absence de référentiels de donnée, élément clé d'un système d'information de type décisionnel réussi
- L'absence de traitement des corrections éditoriales livrées par l'éditeur de la base WoS, Clarivate Analytics.

Tous ces éléments montrent que le système GINKGO est issu d'un projet non terminé et les conséquences sont nombreuses sur sa difficulté d'actualisation et d'exploitation au quotidien.

Data science “old school”

La bibliométrie peut être considérée comme un des nombreux champs d'application des « data science ». Les méthodologies (*data visualisation, machine learning, data mining, ...*) et l'outillage (H2O.ai, SAS, MathWorks, Qlik, ...) en data science ont beaucoup muri et se sont largement démocratisés ces dernières années. On trouve désormais de nombreuses plateformes disponibles sur le marché.

Dans ce contexte l'usage massif de SQL Développer pour extraire les données et R ou Excel pour les analyser semble en décalage avec les pratiques actuelles, d'autant plus vu la complexité par ailleurs du système.

3. LES DIFFERENCES ENTRE TAGGED FILE ET XML FILES

Avant d'analyser les scénarii de passage des données Web of Science du format Tagged files au format XML ce chapitre nous analyse les différences entre ces deux formats.

Structure générale

Une différence importante entre tagged files et XML files est le « niveau » auquel ces données sont attachées (Figure 2).

Dans les tagged files il y a trois niveaux d'information :

- le fichier dans sa globalité
- la section de support éditorial (UI)
- le document (UT)

Dans les XML files toutes les données sont attachées au document (UT), qui est l'unité d'information auxquelles toutes les autres sont rattachées.

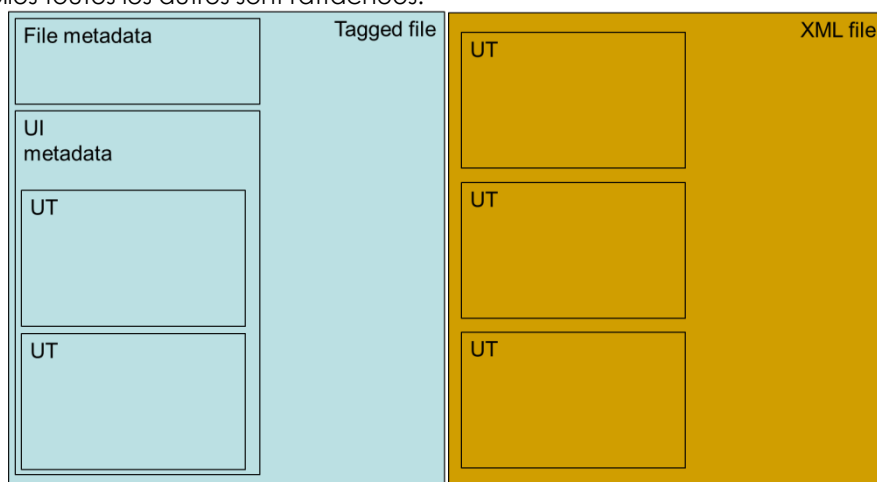


Figure 2 - structure générale d'un « tagged file » (à gauche) et d'un fichier XML (à droite)

Implications

La différence de structure générale a plusieurs implications majeures.

Traitement des fichiers originaux

La machinerie perl permettant d'extraire le contenu des tagged files pour préparer leur intégration dans le DSA ne peut plus être utilisée. Il devient nécessaire de changer la stratégie de chargement du DSA.

Une conséquence corollaire à l'absence de métadonnées avec une atomicité au niveau du fichier est que les statistiques présentes au niveau des tags du fichier ne sont plus présentes dans les fichiers. Il est nécessaire d'utiliser les fichiers Excel de comptage fournis par Clarivate Analytics pour assurer le contrôle qualité des volumes de publication chargés.

Deux types d'approches sont envisageables :

- Dans un scénario conservateur, il paraît raisonnable de remplacer les scripts perl par une mécanique équivalente qui écrirait dans un DSA peu modifié. L'avantage de cette approche est de limiter les risques liés à une refonte.
- Dans un scénario d'évolution, il peut être envisagé de remplacer le DSA par une base de données de type XML Store. L'avantage de cette approche est que la totalité des informations contenues dans les XML seraient stockées, ce qui faciliterait l'exploration du contenu et le retour aux données d'origine en cas de difficulté lors de la validation du chargement.

Disparition du concept d'UI

La principale implication de la différence d'organisation des fichiers est que le concept d'UI, central dans le cas des tagged files, est absent des XML files. La centralité du concept dans les tagged files a conduit à son utilisation centrale dans les schémas Oracle du système GINKGO. Il existe deux approches pour résoudre cette différence :

- recréer artificiellement l'équivalent d'une UI, ce qui permet d'éviter des modifications majeures du système ;
- faire disparaître le concept de support de section éditorial, ce qui permet d'avoir un modèle plus proche de la réalité mais nécessite des changements majeurs du système.

Ces deux solutions seront présentées plus en détail dans la partie consacrée aux différences individuelles entre tagged files et XML files (voir § Clé UI).

Éléments individuels

Certains éléments spécifiques ne sont pas définis de la même manière dans les tagged et les XML files. Ces éléments doivent faire l'objet d'une attention particulière car leur impact est spécifique sur le système GINKGO. Ce chapitre a pour objet de passer en revue les différences spécifiques et leurs impacts. Il proposera pour chacun des solutions à envisager et une recommandation d'application lorsque plusieurs solutions apparaissent envisageables.

Ce qui disparaît

Clé UI

Le concept de clé UI disparaît complètement. L'information « support de section éditorial » est contenue au niveau d'un élément du XML d'un document. Il n'existe plus de clé naturelle.

Impact fonctionnel

Assez important. Le « support de section éditorial » n'est pas un objet d'étude mais certaines normalisations nécessaires au calcul de certains indicateurs se font au niveau de la clé UI. La granularité des études porte au niveau document.

Par ailleurs l'historique de la base (et la centralité de la cleUI dans les schémas Oracle) a conduit à une utilisation courante de la CLEUI dans le travail quotidien.

Impact technique

L'impact technique est potentiellement important. En effet le concept de « category » est rattaché à la clé UI. Ce concept est un des axes d'analyse des rapports produits. Il fait donc l'objet d'une mise en qualité des données, via la réaffectation multidisciplinaire. En conséquence on le trouve dans toute la chaîne de données (DSA, ODS, MDM, QRY).

Solution 1

D'après la documentation des tagged files il existe une combinaison de champs qui peut remplacer la clé ui (Figure 3) :

GA	Document solution identifier	1:1	Var (current max 5 char)	Alphanumeric (upper case)	Identifier for issue or book when combined with year. For unique identifier use 'UI' tag.
-----------	---	------------	-------------------------------------	--------------------------------------	--

Figure 3 - Extrait de la documentation Tagged Files

Le « Document solution identifier » correspond à l'attribut « value » de l'élément <identifier type=« accession_no »> dans les XML files (Figure 4).

```
<identifier type="accession_no" value="BD70T"/>
```

Figure 4 - Exemple d'accession_no

Il est donc possible de remplacer la cléUI par la concaténation « accession_no » + « _ » + « publication_year ».

Cette clé artificielle peut être calculée dès le chargement dans le DSA, afin de minimiser les modifications à effectuer au niveau des schémas en aval. Le type de la colonne cle UI (varchar2(10)) est compatible avec la concaténation « accession_no » + « _ » + « publication_year » (5 + 1 + 4). Avec cette solution il n'apparaît pas nécessaire de modifier ni le DSA, ni les autres schémas pour prendre en compte cette différence.

Avantages :

- modification minimale du processus actuel

Risques :

- présence du champ non garantie

Coût : modéré pour la mise en œuvre.

Solutions 1bis

Historiquement, la cléUI est une sous-chaîne des dix premiers caractères de la cléUT. Il est donc envisageable de calculer une cléUI artificielle par cette méthode.

Avantages :

- modification minimale du processus actuel

Risques :

- Incertitude de la pérennité de la méthode ; en effet, et contrairement à la solution 1, le fait que la cléUI est une sous-chaîne de la cléUT n'est pas documenté.

Coût : modéré pour la mise en œuvre.

Solution 2

Rattachement des catégories directement au document. Faire du concept de « section de support éditorial » un concept secondaire, qui hériterait des attributs du support éditorial ou des documents au besoin.

Avantages :

- Simplification et réalisme des modèles relationnels

Risques :

- Très impactant pour les processus de production de données

Coût : assez élevé pour la mise en œuvre, pas de surcoût pour la suite

Recommandation

Dans un scénario de refonte la **solution 2** est la plus raisonnable car elle aboutit à un modèle qui correspond mieux à la réalité des publications. Le concept de « section de support éditorial » est un concept assez fort (puisque les catégories y sont rattachées) dans le système GINKGO, ce qui ne correspond ni à la réalité éditoriale, ni à la réalité des indicateurs où la catégorisation est un attribut des publications.

Dans un scénario conservateur la **solution 1** est acceptable car elle permet de maintenir le système dans son fonctionnement historique sans prendre de risque important de perte de données. La solution 2 aurait des impacts trop importants pour être envisagée sans une refonte profonde du système.

« SQ » tag (journal / book sequence number)

Le champ "SQ" des tagged files était utilisé comme clé unique naturelle des journaux. Ce champ a disparu.

Impact fonctionnel

Aucun. Les journaux sont généralement identifiés par leur titre.

Impact technique

L'impact technique est très important. En effet le numéro de séquence est la clé primaire naturelle du journal, et est donc utilisée pour la fiabilisation. En conséquence on le trouve dans toute la chaîne de données (DSA, ODS, MDM, DTM).

Solution

La solution la plus sûre est de revoir les modalités de stockage et de fiabilisation des supports éditoriaux. En effet, il n'existe pas de clé naturelle commune et facilement identifiable pour ces supports. L'ISSN est le meilleur candidat de clé naturelle, mais il est parfois absent des données source.

Il serait souhaitable de créer un référentiel de journaux, inexistant dans GINKGO malgré la centralité du concept de périodique dans le contexte de la bibliométrie. L'ISSN pourrait alors être utilisé comme clé primaire dans ce référentiel et le processus de rapprochement entre les données brutes et le référentiel devrait palier au fait qu'il est parfois absent.

Avantages :

- Facilitation de la mise en qualité des données de support éditorial (via notamment la création d'un référentiel)
- Amélioration de la qualité des données
- Amélioration de l'accessibilité aux données
- Utilisation des ISSN (identifiants reconnus)

Risques :

- Très impactant pour les processus de production de données

Coût : assez élevé pour la mise en œuvre dans système actuel, notamment du fait des dépendances actuelles entre différentes tables.

Ce qui change

Références

Dans le cas des tagged files, la référence fonctionne de la manière suivante : une publication citée se voit attribué une clé T9. Cette clé est alors référencée sous le tag R9 dans les publications qui la citent. Dans le cas des XML la clé UT de la publication citée est directement référencée dans les publications qui la citent.

Impact fonctionnel

Aucun. La clé T9 n'est pas utilisée par les chargés d'étude.

Impact technique

Faible. Le DWH et les schémas en aval ne sont pas impactés. L'information à stocker dans le DSA et l'ODS est à modifier ainsi que le processus de remplissage du DWH, mais les concepts sont simplifiés.

Solution 1

Supprimer les clés R9 et T9 des tables où elles sont présentes et remplacer la clé R9 par la clé UT du document cité.

Solution 2

Renseigner le champ existant R9 avec la clé UT citée lors du chargement du DSA.

Recommandation

Dans un scénario conservateur la **Solution 2** est acceptable et très peu couteuse. Dans un scénario d'évolution du système la **Solution 1** est préférable car elle est plus proche de la réalité des données. Ces deux solutions ont un coût assez faible.

Références brevet

Dans les tagged files il existe un ensemble de tags pour décrire les références à un brevet (voir Figure 5).

Tag	Field	Occurrence (Min:Max)	Length	Type	Description
CP	Begin cited patent	1:1	0	Tag only	Begin cited patent sub-record.
/A	Assignee	0:1	Var (current max 75 char)	Alphanumeric (upper case)	Name of assignee of cited patent. Format: {Last name}^(Initials)
/Y	Cited patent year	0:1	Fixed 4 char	Numeric	Publication year of cited patent. Format: YYYY
/W	Cited patent Number	1:1	Var (current max 18 char)	Alphanumeric (upper case)	Patent number or application number.
/N	Cited patent country	0:1	Var	Alphanumeric (upper case)	Abbreviated country of issuance for cited patent. <i>See Chapter Combined Cited Patent.02.12 for a list of abbreviations used.</i>
/C	Cited patent type	0:1	Var (current max 4 char)	Alphanumeric (upper case)	Patent type if not issued patent. Content: APPL (if application), REIS (if reissue).
EC	End cited patent	1:1	0	Tag only	End cited patent sub-record.

Figure 5 - Tags de description des brevets dans "tagged files"

Dans les XML files toutes les références, quelle que soit leur type (article, brevet, livre, ...), apparaissent sous le même élément <reference>.

Impact fonctionnel

Faible. L'information n'est pas utilisée à ce jour.

Impact technique

Faible. Il peut être envisagé plusieurs solutions

Solution 1

Mettre en place une règle de gestion qui sépare les références à des brevets d'un côté ou à d'autres types de documents de l'autre lors du chargement dans le DSA.

Avantage : solution la plus conservatrice, donc la moins risquée ; peu coûteuse.

Solution 2

Ne pas extraire les citations de brevet pour l'instant. Cette solution est envisageable si le DSA est modifié pour contenir les documents XML tels qu'ils sont fournis par Clarivate Analytics.

Avantage : solution évolutive ; coût nul dans le scénario où le DSA stocke le XML

Recommandation

Dans le cas d'un scénario conservateur la **solution 1** est préférable afin de ne pas perdre l'information et devoir retraiter les documents à posteriori.

Dans le cas d'un scénario où les xml seraient stockés dans leur globalité la **solution 2** est recommandée car les données seraient accessibles aisément dans le futur sans qu'il n'y ait de perte fonctionnelle, l'information n'étant pas exploitée à ce jour.

Cardinalité des doctypes

Dans les tagged files la cardinalité des types de documents par rapport aux documents eux-mêmes est 1:1. Un document donné pouvait être présent dans les « proceedings » et dans les « citations index » et potentiellement avoir plusieurs doctype, mais une règle de gestion permettait de choisir lequel garder (en l'occurrence le doctype de SC).

Dans les fichiers XML la cardinalité des doctype est 1:N. Un document peut avoir plusieurs types.

Impact fonctionnel

Potentiellement fort : s'il est choisi de retenir la nouvelle cardinalité il y aura un impact sur certains indicateurs. Il paraît nécessaire de faire une étude exploratoire afin de mesurer cet impact avec plus de précision pour décider la solution choisie (conservation de la cardinalité 1:1 ou utilisation de la nouvelle cardinalité 1:N).

Impact technique

Faible si conservation de la cardinalité 1:1 : il sera nécessaire de mettre en place une règle de gestion au niveau du chargement dans le DSA pour décider l'ordre de préséance des doctype.

Assez fort si passage à la cardinalité 1:N : toute la chaîne de production est impactée, tant au niveau modèle qu'au niveau processus de chargement. Pour la partie modèle, à ce jour le type de document est un attribut de l'objet document à tous les niveaux du système gingko (DSA, ODS et DWH), ce qui n'est pas compatible avec une cardinalité 1:N. Des tables de rattachement devraient être créées. Cela impacterait le processus de chargement lui-même puisque de nouvelles tables seraient à remplir.

Recommandation

Une étude d'impact spécifique semble nécessaire sur ce point afin d'évaluer l'impact au niveau des indicateurs d'un tel changement, et donc sa pertinence.

« Subject » rattachées au document, et plus à l'UI

Il s'agit là d'une conséquence de la suppression de l'information cleUI. Se référer au paragraphe Clé UI.

Contenu de Nomenclature disciplinaire

Dans XML on trouve 2 nomenclatures : « extended » (151 valeurs) & « traditional » (252 valeurs). Dans tagged data on trouve uniquement une nomenclature (traditional).

La nomenclature « traditional » est à utiliser car elle correspond à celle présente dans les tagged files et pour laquelle la table des correspondances est connue avec la nomenclature disciplinaire utilisée dans les rapports.

Parmi les données chargées en 2017 on trouve 251 valeurs sur les 252 déclarées dans la documentation WoS. Parmi ces 251, 250 sont strictement identiques à celles trouvées dans la table GINKGO_DWH.DIM_SPECIALITE. La seule différence est la suivante : 'AUDIOLOGY & SPEAK-LANGUAGE PATHOLOGY' utilisé dans DWH est nommé 'AUDIOLOGY & SPEECH-LANGUAGE PATHOLOGY' dans les XML files.

Impact fonctionnel

Aucun. La nomenclature utilisée dans les rapports, dites nomenclature OST, pourra être utilisée en l'état moyennant une correction de la syntaxe de la spécialité 'AUDIOLOGY & SPEAK-LANGUAGE PATHOLOGY' dans la table de correspondance.

Impact technique

Aucun.

Utilisation de noms complets au lieu de codes

Dans les tagged data plusieurs informations sont codées sur un ou deux caractères (langage, doctype et subject). Dans les fichiers XML les textes entiers sont utilisés, plus les codes.

Impact fonctionnel

Modéré à important. Les rapports produits n'utilisent pas les codes. Cependant les codes sont souvent utilisés par les chargés d'études et statisticiens lors de leurs travaux exploratoires ou leurs validations.

Impact technique

Modéré.

Quel que soit le scénario retenu, c'est une bonne pratique de définir des tables de dictionnaires, qui liste les valeurs possibles. Disposer de ces tables permettrait d'utiliser une approche conservatrice dans laquelle les noms complets seraient remplacés par les codes correspondant lors du chargement dans le DSA. De la sorte le processus de chargement en aval du DSA ne serait pas impacté par cette modification des données sources.

Ce qui apparaît

« Organizations enhanced »

Clarivate Analytics a fait un travail de création d'un dictionnaire d'organisations, appelé Organizations Enhanced. Chaque institution pouvant avoir différentes variantes de son nom ce dictionnaire permet d'indiquer quel est le nom préférentiel de cette institution. Ces informations sont identifiables via l'utilisation d'un attribut pref="Y" inclus dans la balise « organization » quand l'organisation listée provient de l'organization enhanced (Figure 6).

```
<organizations count="2">
<organization>Brigham & Womens Hosp</organization>
<organization pref="Y">"Harvard University"</organization>
</organizations>
```

Figure 6 - Exemple d'organization enhanced

Impact fonctionnel

Fort.

L'utilisation de ce champ permettrait de simplifier le rapprochement institution / adresse en diminuant les variantes dans le nom de l'institution.

Par ailleurs une adresse donnée peut avoir plusieurs organization enhanced. Selon les cas il peut s'agir soit à une multi-affectation de l'adresse, soit d'une hiérarchie d'organisations. Il est nécessaire de pousser l'analyse fonctionnelle sur ce point car l'impact pourrait aller jusqu'aux indicateurs calculés.

Impact technique

Fort. Dans les tagged files la cardinalité adresse – organisation est 1:1. Dans les XML cette cardinalité est 1:N. Il est donc nécessaire de modifier la structure des schémas qui permettent de supporter cette cardinalité.

En fonction de la décision fonctionnelle l'impact pourrait soit couvrir « uniquement » DSA, ODSs et MDM (et leurs processus de chargement), soit la totalité du système, jusqu'aux TSS.

Openaccessness

Clarivate Analytics fournit des données relatives à l'accès « open » ou pas des documents. A ce jour ces données sont fournies via des fichiers XML dédiés. Le sujet est très largement discuté dans la communauté scientifique et le Hcéres souhaiterait valoriser cette information dans certains des rapports qu'il produit.

Impact fonctionnel

Assez fort puisque de nouveaux indicateurs pourraient être introduits.

Une étude plus poussée doit être effectuée pour décider quels indicateurs sont pertinents sur ce sujet.

Impact technique

Potentiellement important.

Dans la mesure où ces données doivent conduire à la création de nouveaux indicateurs l'impact couvre toute la chaîne de production de données. Il est à noter qu'il s'agit d'une information complètement nouvelle, qui n'interfère avec rien d'existant, donc il ne devrait pas y avoir d'impact sur l'existant.

Normalized language

Impact fonctionnel

Le concept de Normalized language est interne à Clarivate Analytics et leur permet une interaction entre différentes bases via un vocabulaire contrôlé. Cette information n'a pas vocation à être intégrée dans la base du Hcéres.

Normalized doctype

Impact fonctionnel

Le concept de Normalized doctype est interne à Clarivate Analytics et leur permet une interaction entre différentes bases via un vocabulaire contrôlé. Cette information n'a pas vocation à être intégrée dans la base du HCERES.

Comptes

Dans le XML, lorsqu'une cardinalité pour un élément est supérieure à 1, le nombre d'occurrences de l'élément dans le document est indiqué dans l'attribut « count » de l'élément parent. La Figure 7 illustre le cas des références d'une publication.

```
<references count="52">
  <reference>
    <uid>WOS:000236942300001</uid>
    <citedAuthor>James, G</citedAuthor>
    <year>2006</year>
    <page>ARTN e32</page>
    <volume>23</volume>
    <citedTitle>A case of self-inflicted craniocerebral penetrating injury</citedTitle>
    <citedWork>EMERGENCY MEDICINE JOURNAL</citedWork>
    <doi>10.1136/emj.2005.032284</doi>
  </reference>
  <reference>
    <uid>WOS:000207858500013</uid>
    <citedAuthor>Chattopadhyay, S</citedAuthor>
    <year>2009</year>
    <page>352</page>
    <volume>16</volume>
    <citedTitle>Fatal transorbital head injury by bicycle brake handle</citedTitle>
    <citedWork>JOURNAL OF FORENSIC AND LEGAL MEDICINE</citedWork>
    <doi>10.1016/j.jflm.2009.01.010</doi>
  </reference>
  <reference>
    <uid>WOS:000298634900003</uid>
```

Figure 7 - Exemple d'attribut "count"

Impact fonctionnel

Avoir accès à cette information permettrait d'avoir une visibilité plus fine sur la complétion des informations d'origine (le nombre d'information présentes dans le fichier correspond-il au nombre d'information annoncées dans le fichier ?) et la complétion du chargement (le nombre d'information présentes dans le DSA correspond-il au nombre d'information présentes dans le fichier ?). La recette technico-fonctionnelle de la base pourrait être facilitée par la présence de ces comptes.

Impact technique

Très faible.

Il est nécessaire d'ajouter une structure de stockage dans le DSA et d'implémenter le processus d'extraction des informations des XML. Cependant l'effort à fournir est très réduit.

Cas des corrections

Clarivate Analytics fournit des mises à jour sur les documents existants (rétractions, corrections d'information). Dans le cas des tagged files les corrections étaient livrées séparément et n'étaient pas chargées dans GINKGO. Dans le cas des XML ces corrections font parties des informations livrées. Par ailleurs, ces corrections sont livrées sous la forme de XML complet, dont on doit considérer qu'il annule et remplace la version précédente.

Impact fonctionnel

Fort.

L'intégration des corrections est un élément fondamental pour la justesse des indicateurs générés.

Impact technique

Fort. D'une part, un certain nombre de dimensions du datawarehouse (par exemple DIM_DOCUMENT ou RAT_CITATION) sont chargées de manière itérative d'une année sur l'autre. La prise en charge des corrections imposerait que tous les modèles soient mis à jour en mode « annule et remplace ». Il est donc nécessaire de s'assurer que les performances de chargement sont compatibles avec ce mode.

4. EVALUATION DU SCENARIO « CONSERVATEUR »

Principes du scénario

Le principe directeur de ce scénario est d'utiliser les fichiers XML comme source de données tout en changeant un minimum d'éléments du système GINKGO. Pour atteindre cet objectif la stratégie retenue est de développer des jobs Talend qui vont extraire l'information des fichiers pour les charger dans un DSA modifié compatible avec le format XML mais le plus similaire possible au DSA servant à charger les tagged files. L'hypothèse est que cette stratégie doit permettre d'utiliser les éléments en aval des DSA sans qu'ils soient impactés.

Le schéma général de chargement, très similaire au système GINKGO actuel, peut être présenté comme dans la Figure 8.

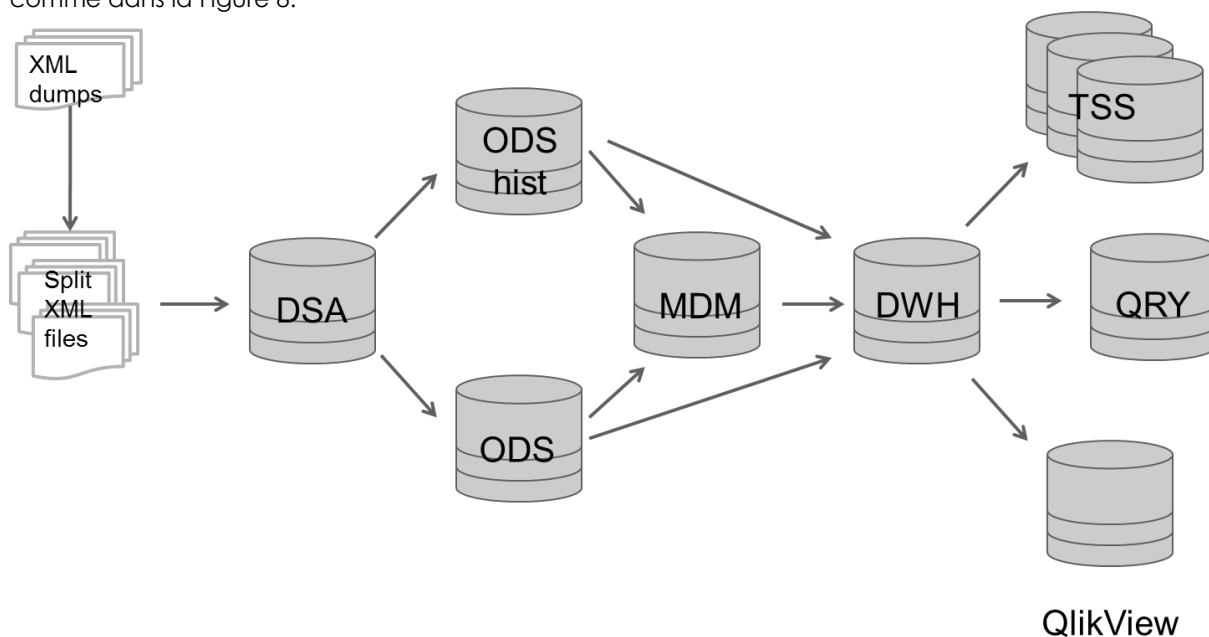


Figure 8 - Schéma général de chargement des données dans le scénario conservateur

Impact du format XML sur le chargement

Le choix du scénario conservateur implique l'approche suivante :

- Découpage des fichiers XML annuels en plus petits fichiers.
- Chargement de ces fichiers via des processus Extract - Transform - Load (ETL).

Il est nécessaire de mettre en place des mécanismes de contrôle des volumes chargés, via l'utilisation des rapports excel fournis par Clarivate Analytics.

Par ailleurs, étant donné la volumétrie importante des fichiers d'origine, il pourrait être nécessaire de mettre en place une orchestration, voire une exécution en parallèle, des processus pour assurer le chargement complet des données. Par orchestration nous entendons une mécanique permettant de savoir à tout moment quelles publications ont été traitées ou pas. L'objectif étant de pouvoir redémarrer le chargement là où il s'est arrêté en cas de problème et d'éviter de devoir redémarrer le chargement depuis le début.

Une étude technique est à mener pour estimer les temps de chargement via cette méthode.

Impact des différences entre tagged files et XML files sur le chargement

Voyons désormais point par point l'impact des différences entre Tagged files et XML files sur le processus de chargement du système GINKGO.

Disparition de la cléUI

Pour rappel, la recommandation dans un scénario conservateur est la Solution 1, p5, et consiste, schématiquement à créer une clé primaire naturelle par concaténation de deux champs.

L'impact est donc principalement au niveau du chargement des fichiers xml dans le DSA, comme illustré dans la Figure 9.

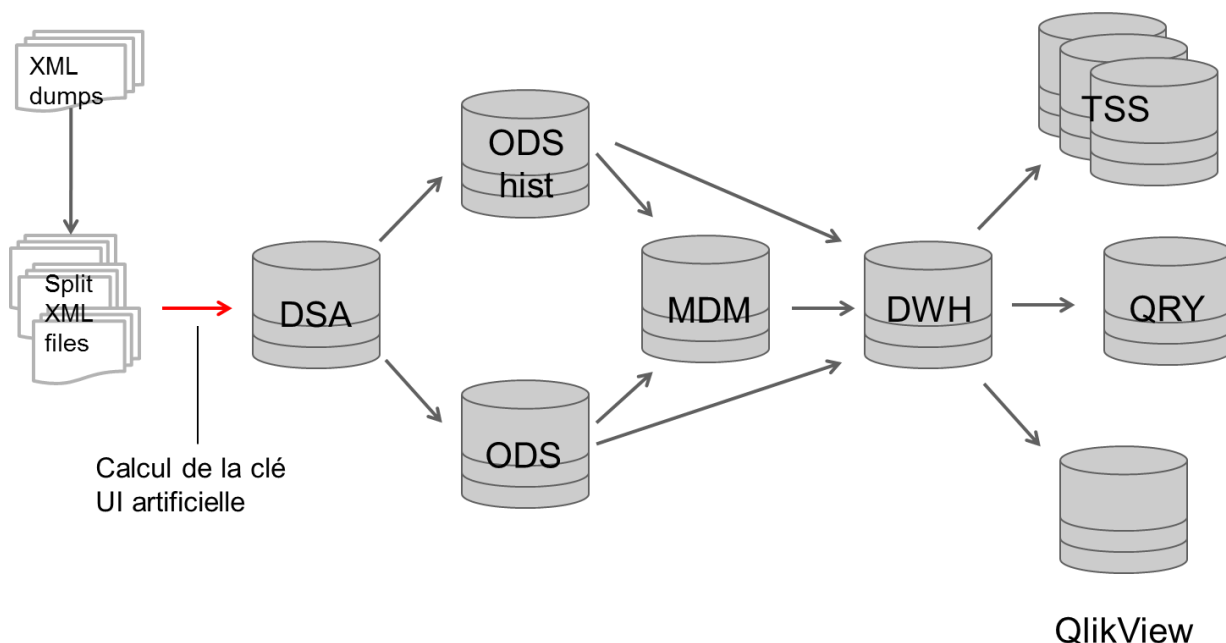


Figure 9 - Impact de la disparition de la clé UI sur le processus de chargement (en rouge)

Disparition du tag « SQ »

Pour rappel la recommandation (cf. Solution, p6) est de revoir les modalités de stockage et de fiabilisation des supports éditoriaux.

L'impact est donc réparti sur l'ensemble des schémas et des processus de mise à jour, comme illustré dans la Figure 10.

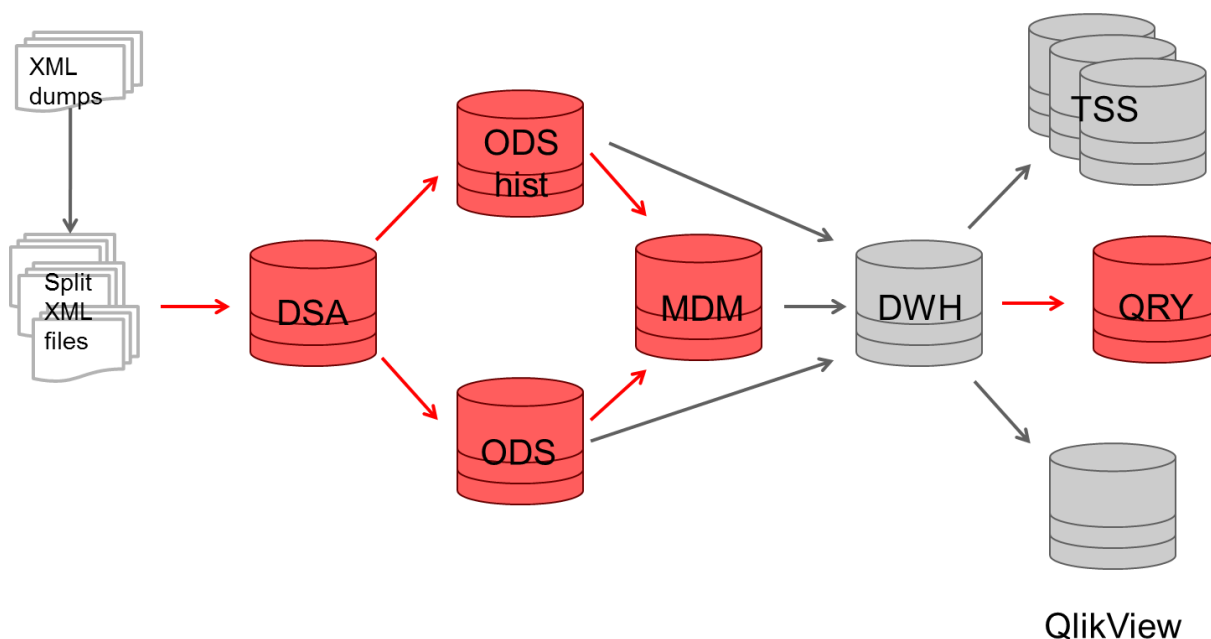


Figure 10 - Impact de la disparition du tag « SQ » dans le processus de chargement (en rouge)

Modification de la déclaration des références

Pour rappel, la recommandation est de renseigner le champ existant R9 avec la clé UT citée lors du chargement du DSA (cf. Solution 2, p7).

Ces impacts sont illustrés dans la Figure 11.

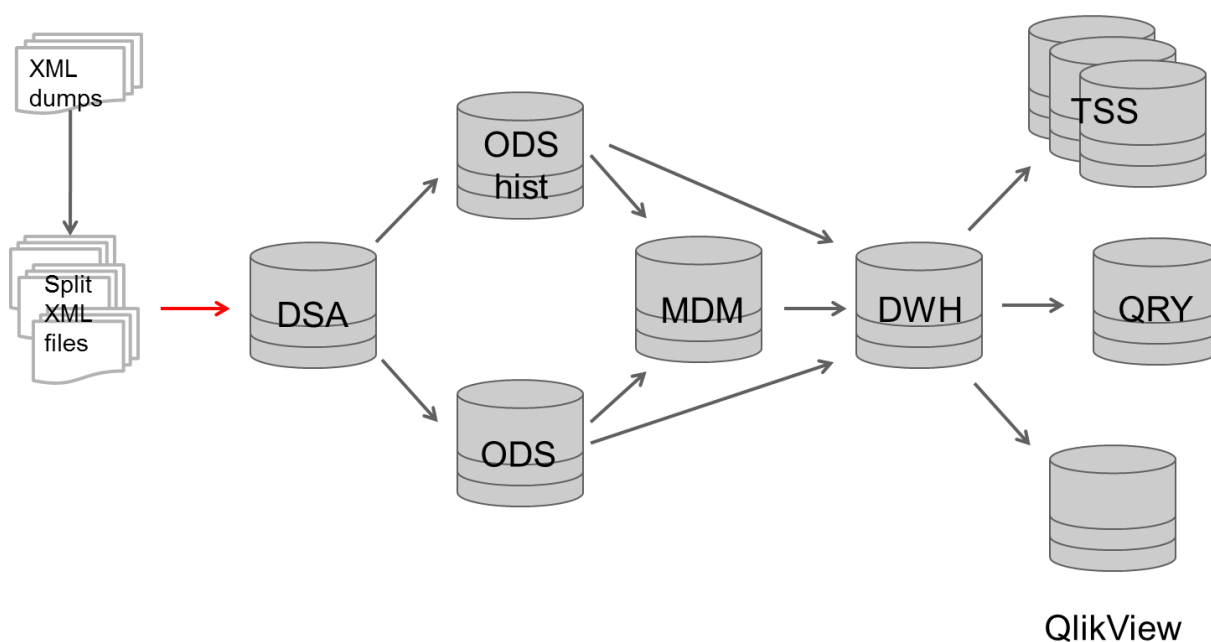


Figure 11 - Impact du changement de déclaration des références sur le processus de chargement (en rouge)

Fusion de la déclaration des références brevet avec les autres références

Pour rappel la recommandation (*Solution 1*, p8) consiste à mettre en place une règle de gestion. L'impact est donc principalement au niveau du chargement des fichiers xml dans le DSA, comme illustré dans la Figure 12.

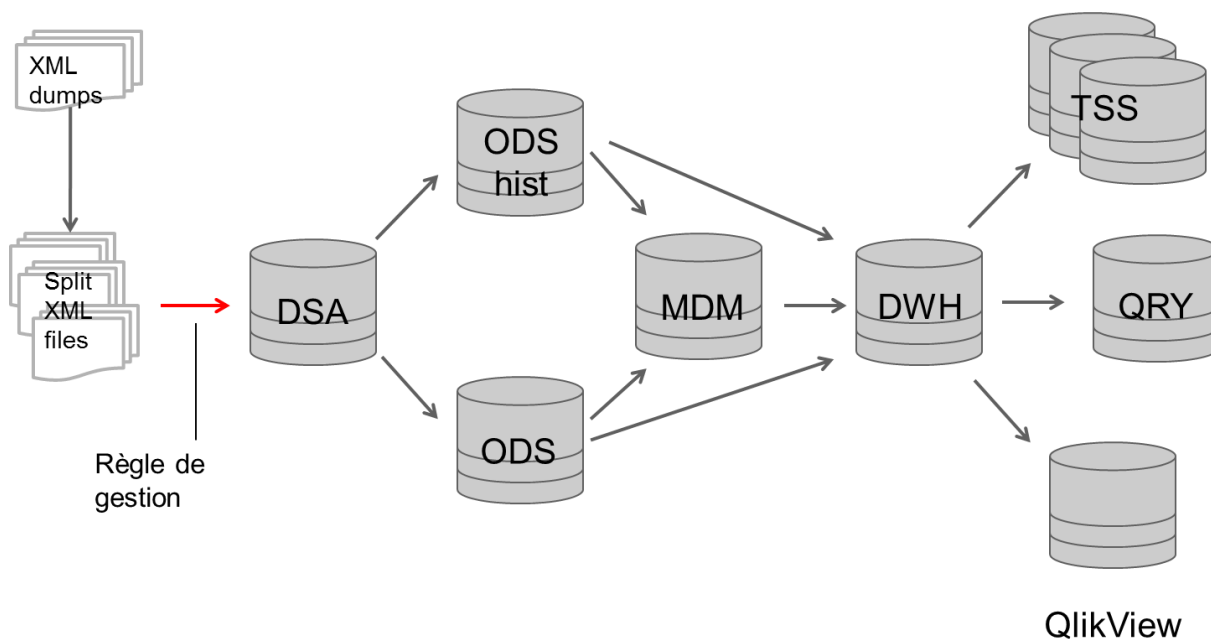


Figure 12 - Impact de la fusion des références à des brevets (en rouge)

Utilisation de noms complets au lieu de codes

Pour rappel, la solution préconisée (cf. Impact technique, p9) est la mise en place de dictionnaires. L'impact est donc d'une part au niveau du MDM, dont l'un des rôles est d'héberger les référentiels, et d'autre part au niveau du chargement du DSA, comme illustré Figure 13.

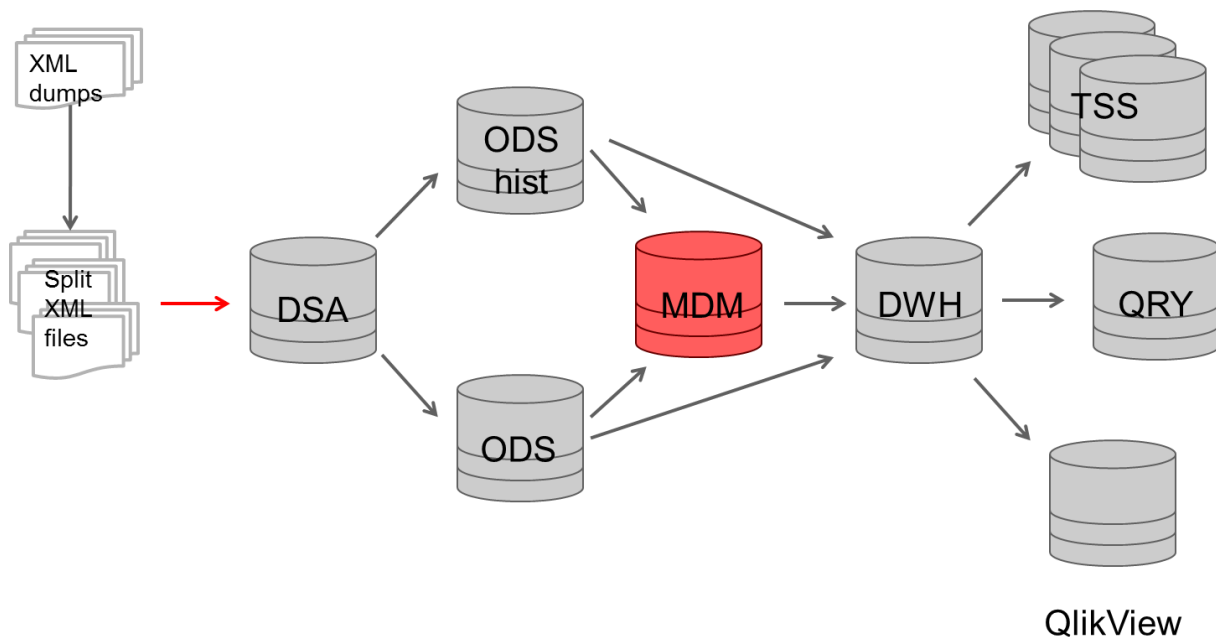


Figure 13 - Impact de la disparition des codes au profit des noms complets (en rouge)

Intégration des Organizations Enhanced

Dans la mesure où une étude fonctionnelle reste à effectuer sur ce sujet il n'est pas possible d'estimer avec certitude la totalité des impacts techniques sur le processus de chargement. Cependant, il apparaît certain que le processus de chargement sera impacté du DSA au MDM. Potentiellement le DWH, les datamarts et la base QlikView seront impactés aussi (voir Figure 14).

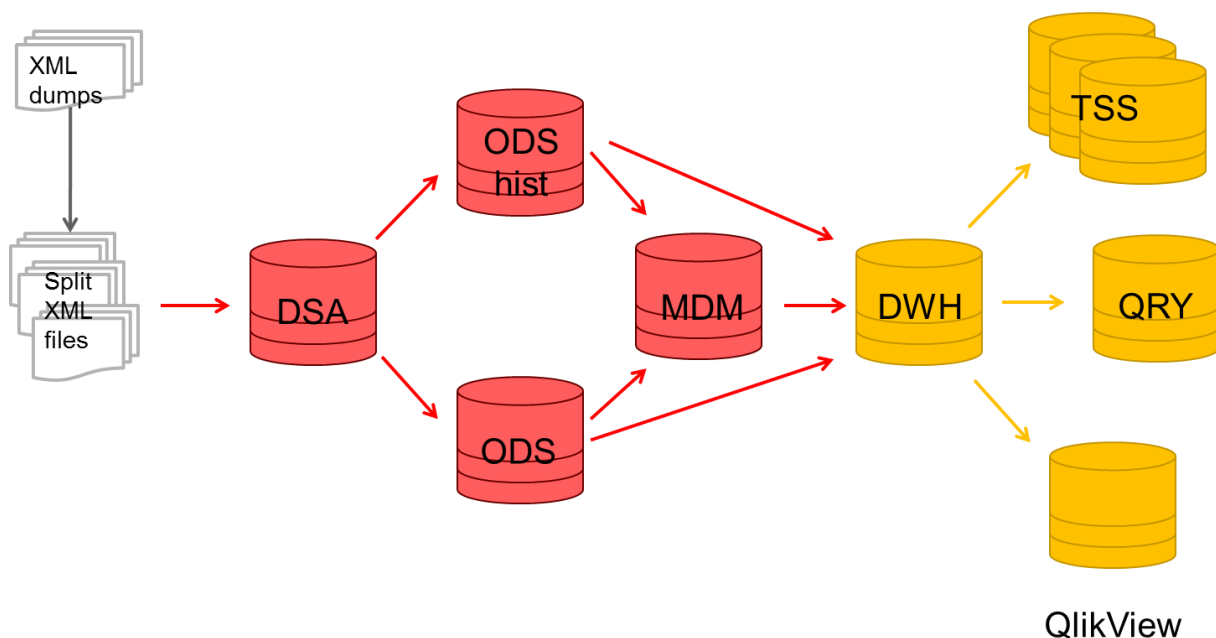


Figure 14 - Impact certains (en rouge) et potentiels (en orange) de l'intégration des "organization enhanced"

Intégration Openaccessness

L'information devant conduire à la mise en place de nouveaux indicateurs elle a un impact sur l'ensemble des éléments du système GINKGO. Par ailleurs, l'information étant pour l'heure transmise via

des fichiers dédiés il sera nécessaire de mettre en place un processus spécifique pour ces fichiers. Ces impacts sont illustrés dans la Figure 15.

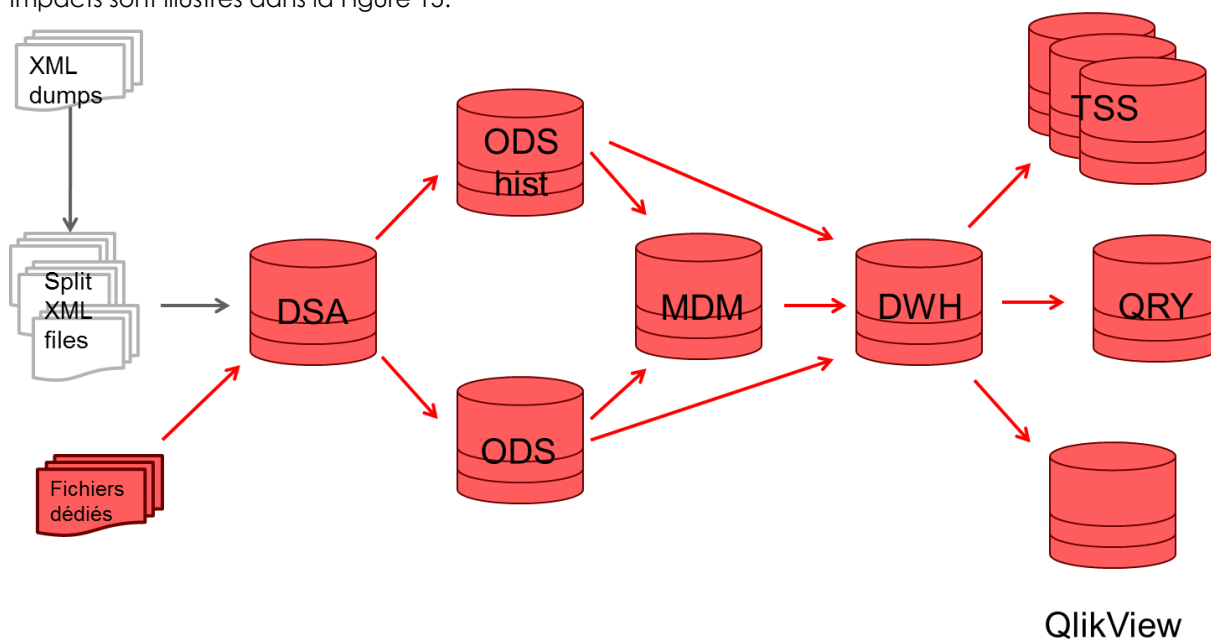


Figure 15 - Impact de l'intégration de l'"openaccessness" sur le chargement des données (en rouge)

Comptes

L'impact de cet ajout est limité aux phases préliminaires du chargement, tel qu'illustré dans la Figure 16.

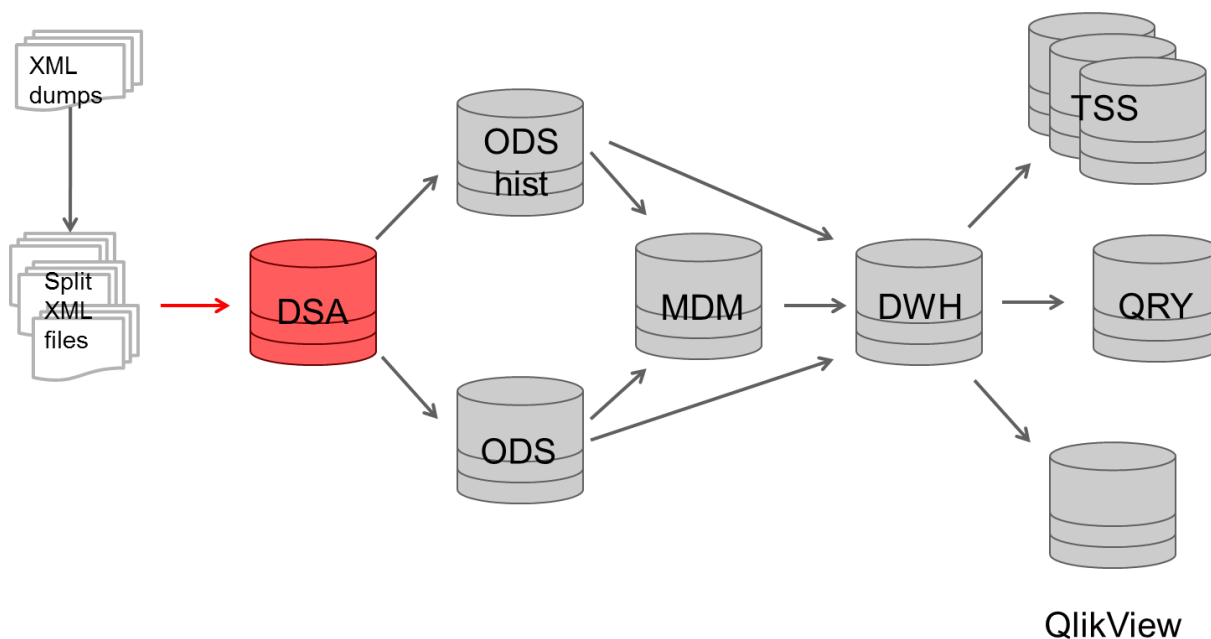


Figure 16 - Impact des comptes d'élément sur le chargement des données (en rouge)

Etat des lieux de l'existant (projet 2017)

Dans cette partie nous analyserons la complétion du projet effectué en 2017 avec les impacts décrits dans ce document.

Pour rappel, l'objectif de ce pilote était d'analyser la faisabilité d'un chargement en base des données XML dans une optique de scénario conservateur. Le scope de ce projet s'est limité à la mise en base dans le DSA. Par nature, tous les impacts situés en aval de ce modèle n'ont pas été pris en charge.

Nous étudierons ce pilote sous deux aspects :

- Le schéma de base de données mis en place
- Les jobs Talend développés

Structure du schema DSA

L'objectif du scénario conservateur étant de modifier le moins d'élément possible nous avons analysé les différences entre le DSA proposé et l'ODS_HIST.

Le Tableau 1 - Liste comparative des tables du DSA et de l'ODS_HIST consigne la liste des tables du DSA et de l'ODS_HIST.

DSA	ODS_HIST
TABLE_STATS	TABLE_STATS
TMP_HEADINGS	
TMP_SUBHEADINGS	
TMP_SUBJECT	
WOS_ABSTRACT	WOS_ABSTRACT
WOS_ADRESSE	WOS_ADRESSE
WOS_ADRESSE_CORRESPONDANCE	WOS_ADRESSE_CORRESPONDANCE
WOS_ADRESSE_NAMES	
WOS_ADRESSE_ORGANISATION	
WOS_ADRESSE_SOUS_ORG	
WOS_ART_DOCTYPE_NORM	
WOS_ARTICLE	WOS_ARTICLE
WOS_ARTICLE_ACK_RCD_GRANT	WOS_ARTICLE_ACK_RCD_GRANT
WOS_ARTICLE_ACK_RECORD	WOS_ARTICLE_ACK_RECORD
WOS_ARTICLE_ACK_TEXT	WOS_ARTICLE_ACK_TEXT
WOS_ARTICLE_AUTEUR	WOS_ARTICLE_AUTEUR
WOS_ARTICLE_DOCTYPE	
WOS_ARTICLE_DOI	WOS_ARTICLE_DOI
WOS_ARTICLE_IDENTIF	WOS_ARTICLE_IDENTIF
WOS_ARTICLE_JNAL_IDENTIF	
WOS_ARTICLE_JOURNAL_TITLE	
WOS_ARTICLE_MOT_CLE_AUTEUR	WOS_ARTICLE_MOT_CLE_AUTEUR
WOS_ARTICLE_MOT_CLE_ISI	WOS_ARTICLE_MOT_CLE_ISI
WOS_ARTICLE_NAMES	
WOS_ARTICLE_ORGANISATION_CORP	WOS_ARTICLE_ORGANISATION
WOS_AUTEUR_ADRESSE	WOS_AUTEUR_ADRESSE
WOS_CATEGORIE_JNAL	WOS_CATEGORIE_JNAL
WOS_CITATION_ARTICLE	WOS_CITATION_ARTICLE
WOS_CITATION_BREVET	WOS_CITATION_BREVET
WOS_CONFERENCE	WOS_CONFERENCE
WOS_CONF_SPONSOR	WOS_CONF_SPONSOR
WOS_CONTRIBUTORS	

WOS_EDITEUR	WOS_EDITEUR
WOS_EDITION	
WOS_INFO_XML	
WOS_JOURNAL	WOS_JOURNAL
WOS_JOURNAL_BS	WOS_JOURNAL_BS
WOS_JOURNAL_BS_BA	WOS_JOURNAL_BS_BA
WOS_JOURNAL_BS_CA	WOS_JOURNAL_BS_CA
WOS_JOURNAL_BS_DN	WOS_JOURNAL_BS_DN
WOS_JOURNAL_BS_ED	WOS_JOURNAL_BS_ED
WOS_JOURNAL_BS_VENTE	WOS_JOURNAL_BS_VENTE
WOS_JOURNAL_EDITEUR	WOS_JOURNAL_EDITEUR
WOS_LANGUE_ARTICLE	WOS_LANGUE_ARTICLE
WOS_LANGUE_ART_NORMALIZED	
WOS_ORG_ADRESSE	WOS_ORG_ADRESSE
WOS_ORGANISATION	
WOS_PRODUIT_SEL_ARTICLE	WOS_PRODUIT_SEL_ARTICLE

Tableau 1 - Liste comparative des tables du DSA et de l'ODS_HIST

Toutes les tables présentes dans l'ODS_HIST sont présentes dans le DSA. Les tables supplémentaires du DSA sont liées à des différences entre tagged files et XML files.

Remarques générales

Certaines colonnes sont de type différent suivant la table dans laquelle elles se trouvent. Par exemple la colonne CLEUT est parfois un VARCHAR2(15) et d'autres fois un VARCHAR2(19).

Le caractère « nullable » ou pas varie selon les tables. De plus il est généralement souhaitable de rigidifier le modèle en limitant les colonnes « nullable ».

Le type VARCHAR2(4000) pour la colonne WOS_ABSTRACT.ABSTRACT devrait être changé en CLOB pour éviter des dépassements potentiels de la limite de 4000 bytes.

Afin d'éviter de limiter les risques d'erreur lors de l'implémentation il semble souhaitable de supprimer les champs qui disparaissent et ne pourront pas être utilisés lors du chargement (WOS_JOURNAL.NUM_SEQUENCE, WOS_ARTICLE.CLEUT9).

Une table a changé de nom (WOS_ARTICLE_ORGANISATION devient WOS_ARTICLE_ORGANISATION_CORP), ce qui impactera le processus de chargement de l'ODS_HIST.

Cas particuliers des différences en tagged et XML files

Disparition de la clé UI

La clé UI a été conservée dans l'ensemble des tables sauf dans la table WOS_CATEGORIE_JNAL. Cependant le champ est toujours vide, la règle de gestion recommandée n'a donc pas été implémentée.

Disparition du tag « SQ »

La colonne NUM_SEQUENCE est toujours présente et elle contient systématiquement la chaîne de caractères 'null'. Ce changement ne semble pas avoir été géré.

Modification de la déclaration des références

La colonne R9 est remplie avec les clés UT. Le cas est donc géré.

Fusion de la déclaration des références brevet avec les autres références

Les citations de brevet sont isolées dans une table dédiée, mais la table est vide. La règle de gestion n'a donc pas été mise en place.

Utilisation de noms complets au lieu de codes

Les codes sont renseignés pour la langue, pas pour le doctype ni pour la category. La solution choisie est donc hybride. Il serait souhaitable d'harmoniser le fonctionnement.

Intégration des Organization Enhanced

L'implémentation de la table WOS_ORGANISATION permet de stocker dans le DSA les différentes organisations (enhanced on non).

Intégration Openaccessness

Cette intégration n'a pas été faite.

Comptes

Les comptes présents dans les fichiers XML n'ont pas été intégrés.

Talend

Des jobs Talend ont été développés pour charger le DSA modifié à partir des fichiers XML. L'objet de ce chapitre est de faire une analyse globale de ces jobs.

La stratégie globale employée a été de développer des jobs élémentaires permettant chacun d'extraire un concept des fichiers d'origine et d'insérer les données dans la table correspondante. La plupart des jobs implémentés lisent la totalité des fichiers XML, extraient les données correspondant à une des tables du DSA, les transforment si besoin, et insèrent ces données dans la table ad hoc. Plusieurs masters jobs ont été développés qui permettent d'orchestrer les différents jobs élémentaires.

Remarque générale

Dans son état actuel le serveur Talend est difficilement utilisable. En effet les temps de manipulation des jobs (ouverture, compilation et sauvegarde notamment) sont extrêmement longs, pouvant causer des attentes de l'ordre de la minute. Ces opérations étant très fréquentes dans un travail de développement, la productivité d'un développeur dans le cadre d'un projet serait fortement impactée.

Jobs élémentaires

Architecture globale

Tous les jobs sont construits sur la même logique, illustrée dans la Figure 17 : tout d'abord les records présents dans la table concernée sont supprimés (truncate), l'activité d'ETL à proprement parler est ensuite effectuée, enfin les statistiques de chargement sont calculées.

Pour la plupart des jobs la partie ETL fonctionne selon la séquence suivante :

- Listing des fichiers à charger
- Lecture des fichiers à charger et extraction des données d'intérêt
- Mapping du contenu du fichier avec les colonnes de la table cible
- Insertion dans la table cible

Il existe quelques jobs, comme par exemple ju_23_alim_art_tmp_article 0.1, qui utilisent comme donnée source des données d'autres tables, et pas les fichiers d'origine.

ju_18_alim_categorie_jnl 0.1	WOS_CATEGORIE_JNAL
ju_19_alim_citation_article 0.1	WOS_CITATION_ARTICLE
ju_20_alim_conference 0.1	WOS_CONFERENCE
ju_21_alimconf_spo 0.1	WOS_CONF_SPONSOR
ju_22_alim_editeur 0.1	WOS_EDITEUR
ju_23_alim_art_tmp_article 0.1	WOS_ARTICLE_JOURNAL_TITLE, WOS_ART_DOCTYPE_NORM, WOS_ARTICLE_JNAL_IDENTIF, WOS_ARTICLE_DOCTYPE
ju_23_alim_article 0.1	WOS_ARTICLE
ju_24_alim_jnal_editeur 0.1	WOS_JOURNAL_EDITEUR
ju_24_alim_jnal 0.1	WOS_JOURNAL
ju_25_alim_jnal_bs 0.1	WOS_JOURNAL_BS
ju_26_alim_jnal_bs_ca 0.1	WOS_JOURNAL_BS_CA
ju_27_alim_jnal_bs_dn 0.1	WOS_JOURNAL_BS_DN
ju_28_alim_jnal_bs_ed 0.1	WOS_JOURNAL_BS_DN
ju_29_alim_jnal_bs_vente 0.1	WOS_JOURNAL_BS_VENTE
ju_30_alim_art_langue_normalized 0.1	WOS_LANGUE_ART_NORMALIZED
ju_30_alim_art_langue 0.1	WOS_LANGUE_ARTICLE
ju_32_alim_wos_article_name 0.1	WOS_ARTICLE_NAMES
ju_33_alim_wos_adresse_names 0.1	WOS_ADRESSE_NAMES
ju_34_alim_wos_adresse_organisation 0.1	WOS_ADRESSE_ORGANISATION
ju_35_alim_wos_adresse_suborganisation 0.1	WOS_ADRESSE_SOUS_ORG
ju_36_alim_wos_contributors 0.1	WOS_CONTRIBUTORS
ju_37_alim_adr_correspondance 0.1	WOS_ADRESSE_CORRESPONDANCE
ju_38_alim_jnal_bs_ba 0.1	WOS_JOURNAL_BS_BA

L'analyse de ces jobs individuels ne montre pas de problème majeur dans l'extraction des données. Cependant on peut noter quelques axes d'amélioration.

Optimisation de la lecture

Chaque job liste et lit l'ensemble des fichiers dans un dossier donné. Si cette stratégie était viable pour le pilote elle ne le sera pas en condition de production. En effet, le temps de lecture des fichiers pour tout le stock de fichier sera très important et la mémoire nécessaire au chargement supérieure à celle disponible.

Capacité de redémarrage

Dans l'implémentation actuelle il n'existe pas de mécanisme de suivi de l'avancée du chargement. Par conséquent, en cas de problème et d'arrêt du chargement il sera nécessaire de recharger la totalité des informations. Du fait de l'importante volumétrie de données à charger la probabilité d'arrêt est extrêmement élevée. Il apparaît nécessaire de mettre en œuvre un mécanisme pour permettre un redémarrage du chargement en cours.

Mutualisation de l'extraction

Plusieurs jobs différents extraient la même portion des fichiers XML (notamment les balises summary/names/name). Il semblerait intéressant de regrouper ces extractions en une seule pour éviter des relectures redondantes des fichiers.

Master jobs

Dans le pilote 4 master jobs ont été développés. L'un, « EXE_Complet_Job » exécute la totalité des jobs élémentaire. Les trois autres (EXE_Alim, EXE_Alim2, EXE_Alim3) s'enchaînent pour appeler les jobs élémentaires.

L'ordre d'appel est légèrement différent entre les deux groupes de master jobs, ce qui n'a pas d'impact sur le chargement général.

Dans le cas où le scénario conservateur serait retenu ces master jobs sont à revoir dans la mesure où ce sont eux qui devraient orchestrer le chargement et ils seront fortement impactés par les remarques d'optimisation faites au paragraphe précédent.

Conclusions sur le scénario conservateur

Le pilote technique a démontré la capacité des outils en place (Oracle et Talend) à extraire des données de fichiers XML.

Il avait déjà montré qu'il ne serait pas raisonnable de mélanger dans le système des données issues de tagged files et de XML files, et que par conséquent il était nécessaire d'envisager un chargement complet du système sur la base des fichiers XML uniquement.

Cette étude montre par ailleurs que le travail pour prendre en charge les différences entre tagged et XML, même dans un scénario conservateur est considérable. En effet, de nombreuses questions ne sont pas réglées, notamment lorsque les impacts des modifications couvrent les parties en aval des ODS. La conclusion sur ce point est que le plus gros du travail reste à faire.

Pour rappel, les éléments issus de la différence entre tagged files et XML et restant à traiter sont les suivants :

- Disparition de la cléUI,
- Disparition du tag « SQ »,
- Fusion de la déclaration des références brevet avec les autres références,
- Utilisation de noms complets au lieu de codes,
- Intégration des Organization Enhanced,
- Intégration des Openaccessness,
- Etude d'impact changement cardinalité des doctypes,
- Intégration des comptes,
- Implémentation Talend

Par ailleurs la nécessité de traiter les corrections livrées au travers des XML files et les volumétries afférentes au rechargement complet du système posent de nouvelles questions sur le processus de chargement et impliqueront très probablement des changements dans les processus de chargement.

Du fait de l'intrication des différents éléments de GINKGO, et de la difficulté de mesurer avec précision les impacts de telle ou telle modification, il est difficile d'estimer le temps de travail nécessaire à l'implémentation du scénario conservateur. Cependant, on peut penser que plusieurs dizaines de jours-hommes seront nécessaires pour prendre en charge toutes les modifications liées au changement de format.

D'autre part, si l'on ajoute à ces considérations le rapport d'étonnement général (voir restitutions partielles et comptes rendus), l'utilisation des XML files comme source des données de publication constitue une opportunité pour le Hcéres de restructurer certains aspects du système GINKGO, notamment le chargement et la mise en qualité des données.

5. SCENARIO EVOLUTIF

La valeur ajoutée de l'Observatoire des Sciences et Techniques du Hcéres repose sur ses compétences en termes de production de rapports d'indicateurs et d'exploration de nouvelles méthodes analytiques. Par conséquent un système d'information au service de ces activités devrait minimiser les tâches de mise à disposition de données fiables, faciliter leur exploration et la production d'indicateurs. Comme nous l'avons vu dans le rapport d'étonnement le système Ginkgo ne sert pas ces objectifs de manière satisfaisante.

Comme nous le verrons dans le paragraphe suivant cette proposition d'évolution se concentre donc principalement sur les « unités » de chargement et de mise en qualité de GINKGO. Par ailleurs, cette proposition vise le maintien en l'état des unités de production d'indicateurs et de reporting automatisé. L'exploration via ExplOST ne devra pas être impactée. L'exploration via requête SQL devrait être simplifiée.

Description générale

L'évolution proposée ici repose sur trois éléments :

- Modification du module de chargement pour stocker les fichiers XML d'origine tels qu'ils sont livrés et exposition des données dans un modèle proche de l'ODS actuel
- Amélioration de la mise en qualité par un travail de constitution de référentiels et de procédure de rapprochement plus robustes

- La création d'un schéma orienté métier qui se substituerait au couple datawarehouse – datamart query.

Cette évolution conduira au schéma proposé en Figure 18.

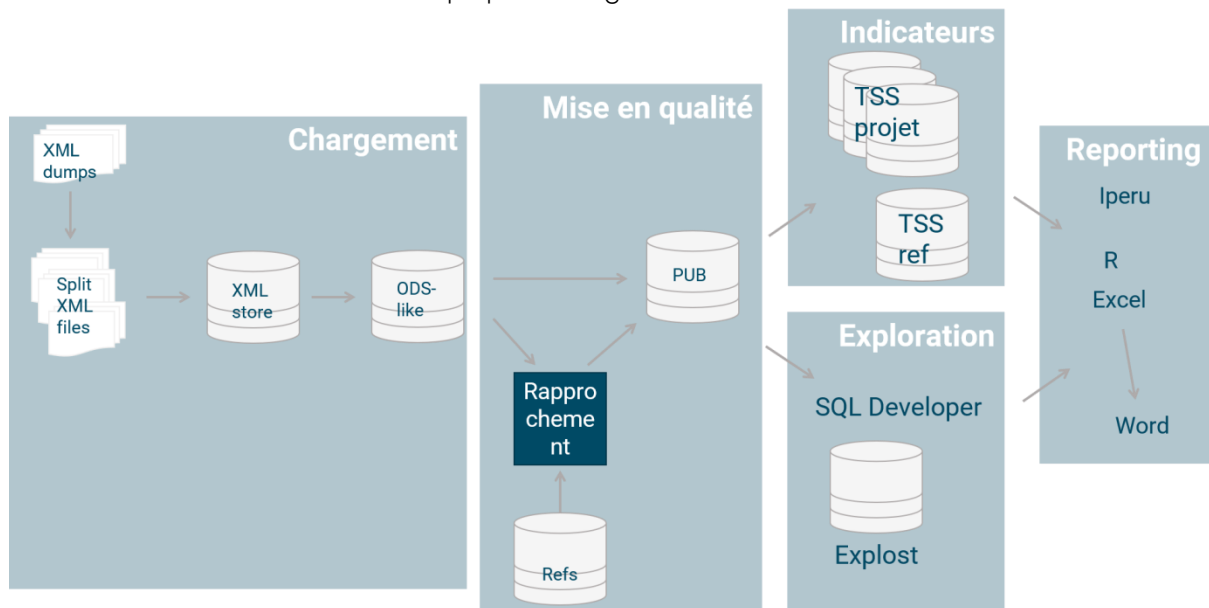


Figure 18- Schéma général après évolution de GINKGO

Chargement

Pour le module de chargement la recommandation est de remplacer le DSA par un « XML store » qui stockerait les documents livrés par Clarivate sans la moindre modification. Cela permettrait de disposer des données d'origine exploitables à tout moment. Pour faciliter l'accès aux publications il est recommandé de stocker un document (au sens publication, actuelle cléUT) par record.

Il existe de nombreux XML Stores sur le marché (Marklogic, Virtuoso, ...). Par ailleurs Oracle propose Oracle XML DB pour cet usage. Il serait tout à fait envisageable d'étudier une option d'achat d'un outil commercial. Cependant, d'une part le besoin actuel est bien couvert par Oracle XMLDB d'autre part un projet d'achat d'un outil peut être relativement chronophage (choix, configuration, appropriation). Cette solution ne semble donc pas pertinente à ce jour. Nous partirons ici du principe qu'Oracle XMLDB sera utilisé pour le chargement.

Plusieurs modes de chargement peuvent être envisagés, comme par exemple la création d'un job Talend ou l'utilisation de SQL*Loader. La solution SQL*Loader est probablement la plus performante et est très simple à mettre en place.

Pour pouvoir fiabiliser les données il sera nécessaire d'exposer les données brutes contenues dans le XML Store. Plusieurs stratégies peuvent être envisagées :

1. Dupliquer les données et les présenter sous la forme de vues matérialisées dans un schéma proche de l'actuel ODS, du schéma Indiana ou de celui de l'OMPI

L'avantage de cette stratégie est qu'elle est simple à mettre en place. Elle permettrait aussi de mettre facilement en place des indexes au besoin. Enfin elle permet d'implémenter des modifications de données si nécessaire, comme par exemple le calcul d'une cléUI artificielle. L'inconvénient est qu'elle duplique les données, ce qui représente un important volume de données. Par ailleurs le temps de mise à jour des vues matérialisées risque d'être important vu les volumes de données.

2. Construire des vues (non-matérialisées)

Le principal avantage de cette stratégie est qu'elle évite de dupliquer les données. Cette stratégie reste simple à mettre en place. Elle permet aussi d'implémenter des modifications de données telles que le calcul d'une cléUI artificielle.

L'inconvénient majeur de cette stratégie est la difficulté technique de mise en place d'indexes. Par conséquent en cas de jointure la performance de requête sur ces vues risque d'être mauvaise.

3. Extraire les données via Talend et ne pas faire à proprement parler de travail d'exposition

Il s'agirait de ne pas créer de vue tabulaire exposant les données XML. Les jobs Talend d'extraction des données et de remplissage des schémas en amont travailleraient directement sur des records au format XML. Cette solution, bien qu'envisageable présente surtout des inconvénients : pas de possibilité d'indexation des données d'origine, imbrication, et donc absence de modularité, de l'exposition des

données brutes et de leur exploitation. Par ailleurs les tests effectués jusqu'ici montrent des problèmes de performance de Talend pour traiter des gros volumes.

L'avantage majeur serait d'éviter une duplication des données.

Recommandation :

Toutes ces solutions requièrent l'écriture de requêtes SQL utilisant XQuery.

La solution #2 est la première à mettre en place car l'écriture des requêtes est la seule et unique chose à faire pour que la stratégie fonctionne. Dans le cas où des jointures s'avèrent nécessaires dans l'avenir et où l'indexation du XML s'avère trop complexe il faudra alors utiliser la stratégie #1. La stratégie #1 peut être considérée comme une évolution de la stratégie #2. Il est donc raisonnable de commencer par tester la stratégie #2, le travail qui sera effectué pour cette mise en place sera quoi qu'il arrive réutilisable.

Mise en qualité

Comme évoqué dans le rapport d'étonnement, la mise en qualité des données est un processus ressenti comme étant coûteux pour l'OST. Par ailleurs il génère de la frustration côté fonctionnel du fait d'une confiance modérée dans la qualité des données produites.

Afin d'améliorer l'efficacité de la mise en qualité des données les recommandations sont de mettre en place

- des référentiels pour les concepts à fiabiliser
- des procédures de rapprochement claires et à jour
- des outils de mesure de la fiabilisation des données

Référentiels et procédures de rapprochement

Dans ce chapitre nous évoquerons les avantages de la création de référentiels et exposeront quelques bonnes pratiques facilitant le succès de ce type de projet.

Avantages de la création de référentiels

L'intérêt de mise en place de référentiels est multiple.

Le premier intérêt est de centraliser de manière explicite les définitions des concepts qui méritent de l'être. Par exemple, dans GINKGO, il n'y a pas de table qui contient de manière normalisée la liste des journaux. Par conséquent, l'extraction des données journaux peut être variable selon la personne effectuant l'extraction. Cela conduit donc à une incertitude sur la qualité des données produites et des temps de validation rallongés pour confronter les résultats et comprendre des différences observées.

L'absence de référentiel empêche de mesurer avec un bon degré de confiance l'écart par rapport à ce qui est considéré comme « la vérité ». Cela conduit donc à des efforts coûteux et incertains pour vérifier la qualité des données. C'est ce que l'on observe dans le cadre de la validation technico-fonctionnelle de GINKGO et plus particulièrement lors de corrections autour de l'enrichissement des adresses.

Par ailleurs, la mise en place d'un référentiel permet de mesurer la qualité de ce référentiel lui-même. Dans le cas du LISST, on peut penser que l'absence de référentiel clair a conduit à une prise de conscience générale tardive de la baisse de qualité.

Disposer d'un référentiel permet de rationaliser une évolution de ce référentiel. Par exemple, les tests autour de nomenclatures disciplinaires alternatives pourraient être facilités par l'existence d'un référentiel. On pourra citer le référentiel NUTS, valable depuis le 1^{er} janvier 2018 et pas encore mis en place, bien qu'il prenne en compte la réforme territoriale française de 2016.

L'existence de référentiels permet par ailleurs la mutualisation entre les systèmes. On peut citer le cas des adresses, qui sont utilisées à la fois dans la base de publications et dans la base de brevets. Ces deux bases bénéficieraient de mutualiser l'effort sur la mise en qualité des adresses.

Enfin, dans le contexte spécifique de GINKGO l'existence de référentiels et l'usage de clés étrangères pour en référencer les entrées permettrait des gains de performance important lors des explorations.

Quelques questions à se poser pour créer des référentiels pertinents

La première question à se poser est celle des concepts pour lesquels la création d'un référentiel serait bénéfique. De manière générale il est pertinent de créer des référentiels pour les concepts dont on veut contrôler la liste, que l'on souhaite partager entre plusieurs systèmes, dont le cycle de vie n'est pas lié à celui des données ou qui constituent des dimensions d'analyse.

Les concepts de GINKGO pour lesquels il semble pertinent de créer des référentiels sont

- les périodiques
- la géographie
- la nomenclature disciplinaire
- les institutions françaises

Modélisation

Pour chaque concept à référencer il est nécessaire de se demander s'il existe et s'il est pertinent d'utiliser un référentiel externe d'autorité (typiquement la nomenclature ISO des pays). Si ce n'est pas le cas il faut modéliser le concept à référencer.

Cycle de vie

Il est nécessaire de décrire à quel moment on considère qu'un nouvel élément est apparu, a disparu ou a été modifié. Il est nécessaire de décrire l'impact de tels événements, sur le référentiel lui-même (historisation ou pas, suppression ou désactivation, ...) et sur les données référençant les entrées concernées.

Les questions autour du cycle de vie doivent s'accompagner de règles de gouvernances claires (mécanismes de validations, responsabilités personnelles).

Afin de faciliter la gouvernance il est important de disposer d'outils de suivi d'événements sur le référentiel (Excel, applications, ...).

Enfin, il peut être intéressant de réfléchir à la possibilité de changer de référentiel de référence s'il existe.

Rapprochement

Pour qu'un référentiel puisse être utilisé il est nécessaire d'établir et de mettre en place des « procédures de rapprochement ». Il s'agit d'algorithmes (potentiellement extrêmement simples), qui vont permettre d'indiquer, avec un certain niveau de confiance, à quelle valeur dans le référentiel correspond la donnée en entrée.

Parmi les questions à se poser figurent la persistance ou pas des rapprochements effectués, le comportement attendu quand une donnée brute échoue à être rapprochée d'une donnée du référentiel, la pertinence du calcul d'un score de confiance au rapprochement.

Par ailleurs Il apparaît important de se doter d'outil permettant d'évaluer la qualité du rapprochement et de l'enrichissement des données brutes.

Avertissement

Lorsqu'on crée des référentiels, le risque est grand d'essayer de définir de manière ontologique les concepts à référencer.

Le travail de définition ontologique est extrêmement chronophage et n'apporte pas de valeur ajoutée importante en l'absence de problématique à proprement parler sémantique (réseau de connaissance, moteurs de recherche, « Internet of Things », ...). Par conséquent il est important de rester pragmatique et de modéliser ce qui arrive, et pas ce qui pourrait arriver, ce qui est utile et pas ce qui pourrait l'être.

En revanche, un référentiel étant amené à évoluer, et éventuellement à répondre à des problématiques en évolution, il est nécessaire de considérer le travail de création de référentiel comme un travail itératif. Il est nécessaire de planifier des revues régulières de référentiel et d'être prêt à le faire évoluer, y compris dans sa structure.

Recommandations générales

C'est une bonne pratique d'éviter le plus possible la duplication des données, et notamment celles de données de référence. Par conséquent, il est fortement conseillé de ne jamais référencer autre chose que ce qui est nécessaire (le plus souvent la clé primaire). Afin d'assurer la lisibilité des données pour les utilisateurs il est conseillé de créer des vues faisant la jointure avec les tables de référence.

Cette pratique a plusieurs conséquences :

- Gain important d'espace disque
- Synchronisme permanent des données exposées aux utilisateurs avec les données de référence
- Centralisation du cycle de vie des données de référence
- Utilisation d'index peu coûteux en maintenance

Référentiels d'intérêt

Dans ce paragraphe nous verrons quels référentiels seraient intéressants à construire et commencerons la description des différents points abordés au paragraphe précédent.

Le référentiel des périodiques

Par périodique on entend les journaux et les séries de livres (« book series ») dans lesquels les documents sont publiés.

Modélisation

Il n'existe pas de référentiel de périodiques d'autorité libre ou déjà souscrit par le Hcéres.

Le concept de périodique est assez simple à modéliser : il existe une clé primaire naturelle qu'est l'ISSN. Un journal a un certain nombre d'attributs : plusieurs titres (titre long, titre 11, titre 21, titre ISO).

Cependant le cycle de vie d'un périodique peut connaître différents événements à prendre en compte dans la modélisation : split, merge, changement de nom mineur (pas de changement d'ISSN), changement de nom majeur (associé à un changement d'ISSN), apparition d'un nouveau support (électronique typiquement). Il est donc nécessaire de créer des relations entre périodiques et peut-être de créer une clé primaire artificielle pour pouvoir regrouper dans une même entité les différents supports d'un périodique donné.

Cycle de vie

Dans le cas des périodiques ce sont les données brutes livrées par Clarivate qui détermineront les éventuelles modifications du référentiel. Par conséquent sa mise à jour sera couplée à l'actualisation du système. Il est probablement souhaitable que la mise à jour du référentiel soit faite après le chargement des données brutes dans le XML Store mais avant la mise en qualité à proprement parler. Ce phasage limiterait le besoin de mise en place de procédures d'aller-retour entre la mise à jour du référentiel et la fiabilisation.

Le concept de périodique n'a pas besoin d'historisation : si un titre change de nom les articles parus dans le journal dans son ancien nom n'ont pas besoin de conserver cet ancien nom. *A contrario* les périodiques n'existant plus (split, merge ou inactivité) doivent être conservés puisqu'il est souhaitable de savoir dans quel journal les articles ont été publiés, même si celui-ci n'existe plus.

Il est à noter que les événements de modification de périodique ne sont pas explicites dans les données ; il est nécessaire de les inférer, et donc probablement de mettre en place des outils de suivi dédié à ces événements. Le suivi de la production par titre est nécessaire et peut-être suffisant.

Il sera nécessaire de déterminer qui a la responsabilité de la mise à jour du référentiel.

Rapprochement

Dans le cas des périodiques le rapprochement peut être basé sur une simple reconnaissance de champs :

- L'ISSN suffira pour une très large proportion de documents
- Quand l'ISSN est absent il sera nécessaire d'utiliser un ou plusieurs champs alternatifs (titre long, titre iso, titre 11, ...)

Si aucune des méthodes précédentes ne fonctionne c'est soit que le titre est nouveau, auquel cas il est nécessaire de mettre à jour le référentiel, soit le que l'information brute présente une erreur. Afin de permettre au système d'apprendre et de traiter ces erreurs il serait intéressant de mettre en place des tables de correspondance de titres alternatifs.

Dans le cas des journaux il n'apparaît pas nécessaire, à ce jour, d'indiquer un score de confiance.

Il est cependant nécessaire d'être capable d'identifier les publications dont les périodiques n'ont pas pu être rapprochés du référentiel. En effet il n'est pas acceptable que certaines publications ne soient pas associées à un périodique. Par ailleurs il est probablement nécessaire de laisser la possibilité de faire des rapprochements manuels, directement sur le schéma d'exposition des données, en gardant la trace de la modification effectuée. Il serait raisonnable de mettre en place une procédure de validation pour ce processus.

Le référentiel géographique

Le Hcéres s'intéresse très majoritairement à 3 niveaux géographiques :

- Les pays dans le monde
- Les grandes régions européennes
- Les communes françaises

Il semble donc pertinent de disposer d'un référentiel pour ces trois niveaux de précision.

Il peut parfois exister des demandes spécifiques sur les niveaux géographiques (états US, villes UK), cependant ces demandes étant ponctuelles et non récurrentes il n'apparaît pas pertinent d'inclure ces granularités dans un référentiel géographique.

Modélisation

Il existe un référentiel d'autorité pour chacun des niveaux géographiques d'intérêt :

- La nomenclature ISO 3166-1 pour les pays du monde
- La nomenclature des unités territoriales statistiques (NUTS) d'Eurostat pour les grandes régions européennes
- Le code officiel géographique de l'INSEE pour les communes françaises

L'effort principal de modélisation a donc déjà été fait par ces trois organismes. Il peut être envisagé de modéliser les relations entre les trois nomenclatures. Par ailleurs, il peut être intéressant de conserver les versions précédentes des nomenclatures, afin de conserver les anciennes attributions géographiques. Cependant ce cas n'est pas fréquent, il est donc possible que sa modélisation soit coûteuse pour une valeur ajoutée limitée. Une approche alternative raisonnable serait de permettre la modification

manuelle et tracée des rapprochements géographiques au niveau du schéma de présentation des données.

Cycle de vie

Le cycle de vie de chacun des référentiels est imposé par la publication des nomenclatures mises à jour.

Une des questions à trancher est de savoir si l'ensemble du schéma de présentation des données est mis à jour pour refléter la nouvelle nomenclature, ou si deux nomenclatures doivent coexister. Prenons l'exemple de la réforme territoriale française de 2016. Les publications antérieures à 2016 associées à l'ancienne région Nord-Pas-de-Calais doivent-elles conserver cette association ou doivent-elles être associées à la nouvelle région Hauts-de-France ?

Par ailleurs les procédures de rapprochement doivent être revues en fonction des évolutions du référentiel, notamment les rapprochements basés sur des reconnaissances de motif (voir § suivant).

Il apparaît nécessaire de mettre en place des procédures, et peut-être des outils, pour étudier l'impact des mises à jour de référentiel. Certains impacts peuvent être négligeables quand ils concernent des régions géographiques à l'extérieur du périmètre principal du Hcéres (typiquement la reconnaissance d'un pays en dehors d'Europe). A l'inverse d'autres modifications peuvent avoir des conséquences plus importantes, comme la fusion de deux communes françaises.

Il sera nécessaire de déterminer qui a la responsabilité de la mise à jour du référentiel.

Rapprochement

Le rapprochement est la problématique la plus complexe pour ce qui est des référentiels géographiques.

En effet il peut y avoir plusieurs problèmes :

- Erreurs dans les données d'origine (discordance ville / pays)
- Ambiguïté des données d'origine (« St Etienne » peut faire référence à « Saint Etienne du Rouvray »)
- Graphies alternatives (St., Saint, St)
- Erreurs typographiques (Pariis)
- Langues alternatives (Parijs)

Par ailleurs il existe des règles de gestion spécifiques dans certains cas : bien que les DOM-TOM français possèdent chacun un code ISO propre on leur attribuera le code ISO de la France.

Il est donc nécessaire de décrire et d'implémenter un algorithme de rapprochement. L'approche la plus simple consisterait à créer un arbre de décision. Une approche plus complexe consisterait à implémenter des mécanismes probabilistes pour effectuer le rapprochement. Il paraîtrait raisonnable de traiter tous les cas sans ambiguïté par un arbre de décision et les cas ambigus par une approche probabiliste. Dans ce cas serait intéressant de mettre en place des outils pour permettre à des humains de trancher en cas de score de confiance insuffisant.

Par ailleurs il serait important de mettre en place des procédures de validation des rapprochements qui permettent d'évaluer à la fois l'intensité du rapprochement et sa justesse. Il est à noter que les niveaux de qualité attendus ne sont pas les mêmes pour les différentes granularités de géographie.

Enfin, il est probablement nécessaire de laisser la possibilité de faire des rapprochements manuels, directement sur le schéma d'exposition des données.

Le sujet des référentiels géographiques et des problématiques de rapprochement fait actuellement l'objet d'une étude dédiée.

Le référentiel des institutions françaises

Dans la mesure où les institutions françaises constituent un axe d'étude important pour le Hcéres il serait bénéfique de disposer de la liste administrée des institutions étudiées.

Modélisation

Il n'existe pas de référentiel d'autorité des institutions françaises.

Le besoin premier est de disposer de la liste des institutions étudiées. Par conséquent un modèle très simple basé sur une seule table peut suffire.

Cycle de vie

Dans le cas des institutions c'est l'actualité académique qui détermine d'éventuelles modifications du référentiel. Ces événements sont relativement fréquents. Cependant, le nombre total d'institutions à prendre en compte est relativement faible (cf institutions IPERU). Il est donc raisonnable de les traiter au cas par cas, et donc de façon manuelle. L'impact sur les données est lui-aussi à traiter au cas par cas.

Il sera nécessaire de déterminer qui a la responsabilité de la mise à jour du référentiel.

Rapprochement

Le rapprochement entre les données et le référentiel institutionnel est un travail complexe, notamment du fait que la manière de renseigner l'institution dans les données source est très variable (présence ou pas de l'équipe, présence ou pas du groupe, tutelles multiples, pas de nomenclature de nommage ...). La mise en place d'un algorithme avancé de rapprochement pourrait être envisagée afin de faciliter la mise en qualité des données institutionnelles. Cependant, la stratégie actuellement utilisée est celle du rapprochement institutionnel, c'est-à-dire que les institutions sont chargées, via une interface dédiée, d'identifier (ou de valider) les publications qui leur sont rattachées. L'optimisation de ce processus est en dehors du scope de cette étude.

Le référentiel de nomenclature disciplinaire

Modélisation

La nomenclature disciplinaire de l'OST est une taxonomie à 2 niveaux (domaine et grande discipline). C'est une nomenclature « maison », basée sur la nomenclature des « catcodes » fournie par Clarivate Analytics.

Si la nomenclature des catcodes est très utilisée, il n'existe pas de référentiel de nomenclature disciplinaire d'autorité, par exemple fourni par une institution mondiale. Le Hcéres teste actuellement la nomenclature « ERC » (European Research Council), qui a la même structure de taxonomie à 2 niveaux.

Cycle de vie

La nomenclature ERC et la nomenclature des catcodes subissent régulièrement des révisions avec ajout ou suppression de catégories. L'impact de ces modifications sur les données devra être discuté lors de l'établissement du référentiel.

Rapprochement

Le rapprochement entre les données et le référentiel est très simple et fait par l'intermédiaire de tables de correspondance entre les catégories présentes dans les données d'origine et le niveau « grande discipline » de la nomenclature.

Tables de correspondance

Lors de l'analyse des différences entre Tagged Files et XML il est apparu que la langue, le type de document et la nomenclature disciplinaire ne sont plus référencés en tant que codes (1, 2 ou 3 caractères) mais en toutes lettres. Afin d'assurer la continuité technique de la production d'indicateurs et de minimiser les efforts de gestion du changement au niveau de l'exploration, il est préconisé de créer des tables de correspondance pour les trois concepts cités précédemment afin de disposer des codes dès la couche d'exposition des données du XML Store.

La préconisation de créer des vues au niveau du XML est tout à fait en accord avec la constitution de tables de correspondance.

Mise en œuvre de l'outillage

Après avoir passé en revue les référentiels d'intérêt pour le Hcéres, voyons comment mettre en œuvre ces référentiels.

Du point de vue l'outillage il est possible d'envisager trois stratégies pour la mise en œuvre des référentiels :

- Mise en œuvre manuelle (SQL et excel comme outils principaux) des processus décidés
- Achat d'une solution commerciale
- Développement d'un outil

La solution manuelle est celle employée à l'heure actuelle pour la mise en qualité des données. C'est la plus rapide et la plus simple à mettre en œuvre. Cependant, dans la mesure où elle est celle qui a cours actuellement, on peut penser qu'elle n'est pas complètement étrangère au coût ressenti pour l'actualisation.

De nombreux éditeurs possèdent des outils de « master data management » (MDM) tels qu'Oracle, IBM, Talend, SAS. L'avantage de ces solutions est leur maturité et donc le fait que la plupart des règles de gestion envisagées pourront être configurées. Les risques de ces solutions sont un surdimensionnement (seul un petit nombre de fonctionnalités sera utile au Hcéres), un coût de mise en œuvre élevé (notamment de licence) et une certaine longueur de mise en place (choix de solution, installation, configuration, intégration à l'existant ; ce genre de projet s'étale généralement sur plusieurs mois). Dans la mesure où le Hcéres possède déjà Talend il pourrait être intéressant d'envisager cette solution.

Le développement d'outils internes paraît pertinent car il permettrait d'améliorer la mise en qualité tout en capitalisant sur les technologies maîtrisées par le Hcéres : Oracle pour la partie persistance et Talend pour les algorithmes de rapprochement. Pour les aspects d'administration des référentiels et de suivi des indicateurs de la mise en qualité une application web pourrait être développée à moindre coût grâce au framework de développement Application Express d'Oracle (voir Figure 19). Un développement à façon permettrait de disposer d'outils adaptés aux besoins spécifiques du Hcéres.

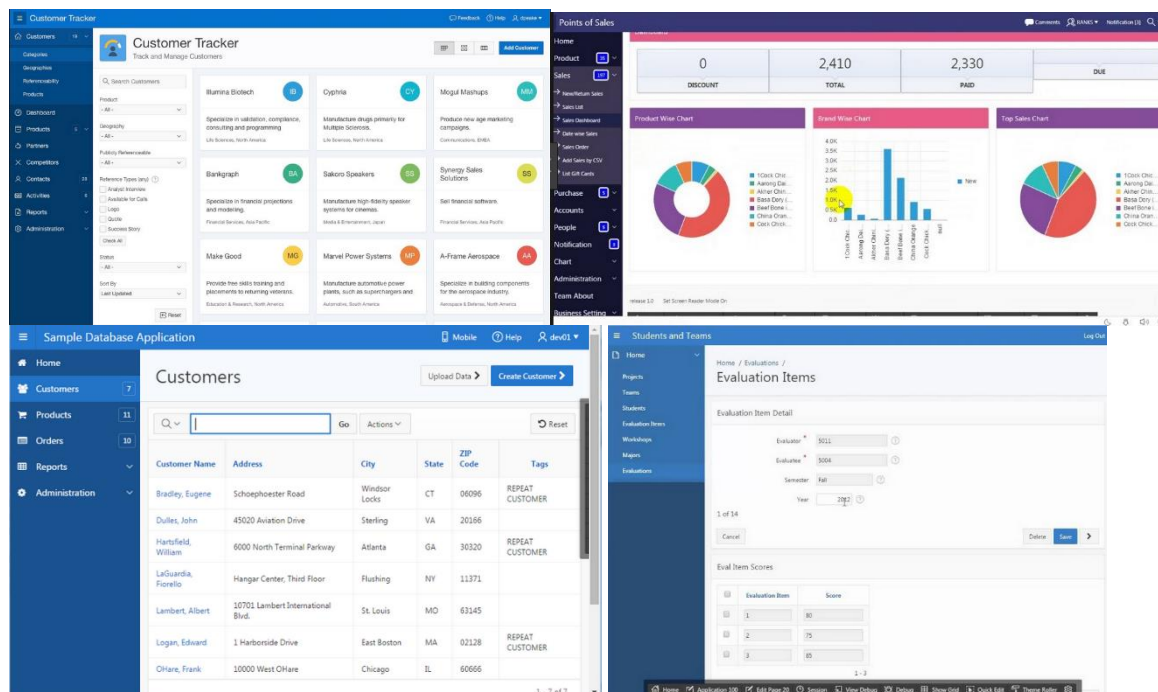


Figure 19- Quelques exemples d'interfaces développées grâce à Oracle Application Express

Création d'un schéma orienté métier

Comme indiqué dans la Description générale de la proposition d'évolution du système, le troisième élément central est la création d'un schéma orienté métier qui se substituerait au couple datawarehouse – datamart query.

Pour rappel, le datawarehouse répondait au besoin de stockage de données multi-sources. Le datamart query, quant à lui, était le schéma initialement dédié à l'exploration des données.

Etat des lieux

L'application des règles de l'informatique décisionnelle à la conception a conduit à produire le datawarehouse et le datamart query. En effet, dans la logique décisionnelle il existe un schéma, le datawarehouse, qui sert à entreposer les données normalisées et standardisées issues des multiples bases agrégées. Ce schéma n'est pas dédié à être consulté par les utilisateurs.

La consultation des données par les utilisateurs se fait à travers des datamarts, qui contiennent les données utiles au type d'utilisateur ciblé. Dans un système d'informatique décisionnelle classique il existe plusieurs datamarts répondant chacun aux besoins de certains profils d'utilisateurs.

Cette configuration est intéressante lorsqu'il existe plusieurs types de données à la source et plusieurs types d'utilisateurs n'ayant pas besoin des mêmes données. La contrepartie de cette configuration est la complexité, notamment de la manipulation des données (normalisation / dénormalisations successives) et des structures de bases de données, qui doivent être suffisamment génériques pour stocker des données assez différentes.

Dans le contexte de GINKGO la motivation de cette logique était la volonté de stocker des données de diverses sources bibliographiques ou bases brevet. Cet objectif n'a pas été atteint lors de la mise en place du système. Par ailleurs cet objectif n'est plus d'actualité. On constate en effet qu'une base brevet totalement indépendante a été mise en place.

Voici quelques exemples de complexification amenées par GINKGO et qui ne répondent à aucun besoin utilisateur actuel :

- Changement de nom des colonnes (cleUT devient id_document, cleui devient id_section_support_editorial)

- Démultiplication de l'information, et donc difficulté d'identification de l'information pertinente (cas des journaux)
- Complexité de certaines modélisations. Par exemple le concept d'entité géographique est un concept hiérarchique alors que la réalité du besoin est un triplet ville (pour la France), NUTS (pour l'Europe), pays (pour tous) ; par ailleurs la vue qui permet d'exposer simplement ce triplet n'existe pas.
- Présence de nombreuses tables (par exemple ACTU_* dans DTM_QRY), visibles par certains utilisateurs et pas par d'autres. Lors de l'accès aux données cela crée de la confusion chez les utilisateurs
- Présence de concepts inutilisés (auteur, hiérarchie des structures de recherche)

Sur le plan technique, on pourra reprocher différents aspects au système actuel :

- De très nombreuses données sont répétées dans de très nombreux endroits ; l'espace disque utilisé par GINKGO est très largement supérieur à ce qu'il pourrait être
- Absence de clés primaires de certaines tables (par exemple dans le warehouse DIM_ADRESSE, DIM_ENTITE_GEOGRAPHIQUE, DIM_REGROUPEMENT, DIM_SPECIALITE, DIM_SUPPORT_EDITORIAL, DIM_TITRE_SUPPORT_EDITORIAL, DIM_TYPE_REGROUPEMENT, RAT_ADR_AUT_DOC_HIE_STR, RAT_SPE_DOC_SEC_SUPP_EDI), et désactivation dans d'autres (DIM_AUTEUR, RAT_DOCUMENT_MOT_CLE)
- Absence de clés étrangères bien que des clés primaires soient référencées (voir Tableau 3 pour le warehouse)

Tableau 3 - Lise des tables du datawarehouse référençant une clé primaire sans qu'elle soit déclarée comme clé étrangère

Table	Colonne	Table contenant la clé primaire
DIM_DOCUMENT	ID_SECTION_SUPPORT_EDITORIAL	DIM_SECTION_SUPPORT_EDITORIAL
DIM_DOCUMENT	ID_TYPE_DOCUMENT	DIM_TYPE_DOCUMENT
DIM_ENTITE_GEOGRAPHIQUE	ID_TYPE_ENTITE_GEOGRAPHIQUE	DIM_TYPE_ENTITE_GEOGRAPHIQUE
DIM_REMERCIEMENT	ID_DOCUMENT	DIM_DOCUMENT
DIM_RESUME	ID_DOCUMENT	DIM_DOCUMENT
RAT_ADR_AUT_DOC_HIE_STR	ID_AUTEUR	DIM_AUTEUR
RAT_ADR_AUT_DOC_HIE_STR	ID_DOCUMENT	DIM_DOCUMENT
RAT_ADR_AUT_DOC_HIE_STR	ID_HIERARCHIE	DIM_HIERARCHIE
RAT_DOCUMENT_MOT_CLE	ID_DOCUMENT	DIM_DOCUMENT
RAT_DOCUMENT_MOT_CLE	ID_TYPE_MOT_CLE	DIM_TYPE_MOT_CLE
RAT_EDITEUR_SUPPORT_EDITORIAL	ID_EDITEUR	DIM_EDITEUR
RAT_LANGUE_DOCUMENT	ID_DOCUMENT	DIM_DOCUMENT
RAT_SPE_DOC_SEC_SUPP_EDI	ID_DOCUMENT	DIM_DOCUMENT
RAT_SPE_DOC_SEC_SUPP_EDI	ID_SECTION_SUPPORT_EDITORIAL	DIM_SECTION_SUPPORT_EDITORIAL
RAT_STRUCTURE_HIERARCHIE	ID_HIERARCHIE	DIM_HIERARCHIE
RAT_STRUCTURE_HIERARCHIE	ID_STRUCTURE	DIM_STRUCTURE
RAT_STRUCTURE_HIERARCHIE	ID_TYPE_STRUCTURE	DIM_TYPE_STRUCTURE
RAT_SUPPORT_SECTION_TITRE	ID_SECTION_SUPPORT_EDITORIAL	DIM_SECTION_SUPPORT_EDITORIAL
SNAP_UPDATE_DOCUMENT	ID_DOCUMENT	DIM_DOCUMENT

Pour le datamart query, on constate dans la pratique que les chargés d'étude et les statisticiens utilisent extrêmement souvent d'autres schémas que le datamart query. Cela tend à montrer qu'il ne répond plus aux besoins d'exploration rencontrés par les « fonctionnels ». L'ODS_HIST et le MDM sont très souvent utilisés en complément du datamart query.

Recommandations générales

L'utilisation d'un modèle en flocon est certainement le modèle de conception de base de données le plus adapté pour modéliser les données bibliographiques.

Les concepts utiles à ce jour à l'exploration et à la production des indicateurs sont les suivants : document, journal et UI, institutions françaises, localisation (ville française / région européenne / pays) et discipline. Ces concepts et leurs relations peuvent être schématisés comme dans la Figure 20.



Figure 20 - Concepts utiles aux mesures d'impact des publications de l'OST

Il n'y a pas de raison identifiée à ce jour pour faire un modèle contenant plus d'entité.

On remarquera que dans ce modèle un grand nombre d'éléments sont en réalité des objets décrits dans des référentiels (institutions françaises, périodiques, disciplines et localisation). Par conséquent les données n'ont pas besoin d'être dupliquées, seules les clés primaires des éléments des référentiels sont réellement nécessaires et pourront être référencées par des clés étrangères dans des tables de rattachement. Cette approche présente plusieurs avantages majeurs : un gain d'espace considérable, une plus grande robustesse du modèle (grâce aux contraintes d'intégrité), une performance de requêtage accrue (indexes sur les clés étrangères et les colonnes des référentiels). La simplification du modèle et la diminution des volumes de données à écrire permettra de compenser la perte de performance à l'écriture inhérente à l'utilisation d'indexes et de contraintes d'intégrité.

Afin de rendre ce modèle exploitable par les utilisateurs il est nécessaire de créer des vues (non-matérialisées) effectuant les jointures entre les tables du modèle proposé ici, ses tables de rattachement et les tables (ou vues si la modélisation de l'objet référencé le nécessite) des référentiels. Les vues sont un outil très puissant et complètement ignoré par GINKGO. Elles permettent d'exposer des données pertinentes aux utilisateurs et évitent de dupliquer les données physiquement. Couplées à des indexes, notamment sur les clés étrangères, elles sont très performantes.

Les quatre tables de fait du datamart query (FAIT_DOC_GEO, FAIT_DOCUMENT, FAIT_FENETRE_CITATION_RECUES et FAIT_SECTION_SUPPORT_EDITORIAL) étant nécessaires à la génération des indicateurs elles devront être conservées. Elles peuvent être reproduites à partir des données disponibles dans le schéma présenté en Figure 20. Ces tables peuvent être créées soit sous la forme de table, soit sous la forme de vues matérialisées. Pour des raisons d'optimisation d'espace et de performance il est recommandé de créer des tables de faits contenant uniquement les identifiants nécessaires (clés primaires référencées) et les colonnes calculées. Pour être exploitables par les fonctionnels elles pourront être exposées sous la forme de vues.

Enfin, une dernière recommandation générale est d'exposer aux utilisateurs uniquement les données dont ils ont besoin. On a vu que la présence des tables ACTU_*, NWOS_* dans le datamart query, par exemple, et le fait que tous les utilisateurs ne voient pas les mêmes données le rend difficilement compréhensible. Pour éviter ce phénomène plusieurs bonnes pratiques peuvent être mises en place :

- créer des schémas « techniques » si nécessaire si des tables intermédiaires sont indispensables
- utiliser le concept de vue (non-matérialisée) et donner accès aux utilisateurs uniquement à ces vues (l'utilisation de profils Oracle permet de gérer finement mais globalement les droits d'accès de chaque utilisateur Oracle)

Impact sur les parties aval du système

Dans ce chapitre nous analyserons l'impact du changement proposé sur la création des TSS d'une part et de la base sous-jacente à Qlikview d'autre part.

Impact pour les TSS

Le flux de création des TSS est illustré dans la Figure 21.



Figure 21 - flux de création des TSS

Par conséquent, s'il est possible de reproduire soit les tables ACTU_*, soit les tables NWOS_* et {TA|TB}_* la modification du datawarehouse n'aura aucune conséquence sur la faisabilité des TSS. Le contenu des tables NWOS_* et {TA|TB}_* est présenté dans le tableau ci-dessous.

Tableau 4 - Liste des colonnes dans les tables NWOS_* et {TA|TB}_*, nécessaires à la création des TSS

NWOS_SA	NWOS_TA	NWOS_TB	NWOS_TC	TA_ANNEE	TB_ANNEE
JNAL	DOCINDEX	DOCINDEX	DOCINDEX	DOCINDEX	DOCINDEX
PRODCODE	DBYEAR	DBYEAR	DBYEAR	ANNEE	ANNEE
CATCODE	YEAR	YEAR	YEAR	DBYEAR	DBYEAR
ANNEE_PUBLI	JNAL	NIVEAU3	FENETRE	JNAL	PAYS
DATE_EN_BASE	CLE_JNAL	CODE_ISO	CITATION	CLE_JNAL	NIVEAU3
CLE_JNAL	LANG	CODE_NUTS		LANG	NIVEAU2
	DOCTYPE	NBCTRYOC		DOCTYPE	NIVEAU1
	VOL				NIVEAU11
	PAGE				NIVEAU12
	AUTEUR				NIVEAU13
	WOS_AUTEUR				NIVEAU14
					NIVEAU15
					NIVEAU16
					NUTS3
					NUTS2
					NUTS1
					NIVEAU0
					NBCTRYOC

L'analyse de ces tables montre qu'il existe plusieurs types de champs :

- les champs en vert pourront être présents dans le modèle proposé puisqu'ils sont soit des entités (JNAL, DOCINDEX), soit des attributs d'entités (LANG, DOCTYPE, DBYEAR, YEAR).
- les champs « NIVEAU » correspondent à un système de codification des localisations géographiques ; ils devront donc être présent dans le référentiel géographique pour assurer la continuité de la production des indicateurs
- les champs en orange sont des champs calculés et les données sous-jacentes sont présentes donc ces calculs ne poseront pas de problème de faisabilité
- les champs AUTEUR et WOS_AUTEUR sont en réalité inutilisés, ils peuvent être remplacés par des données fictives pour le maintien de la structure des tables

En conséquence, la substitution des datawarehouse & datamart query par un schéma orienté métier n'impacte pas la faisabilité du calcul des TSS. L'impact sera au niveau de la réécriture des scripts conduisant à la création des quatre tables.

Impact pour Qlikview

L'impact sur la préparation des données pour les applications Qlikview n'a pas été estimé en détail (hors du périmètre de la mission).

Cependant, si l'on prend comme exemple le cas de l'application ExplOST, son modèle de données s'appuie sur 20 tables de GINKGO, comme illustré dans la Figure 22.

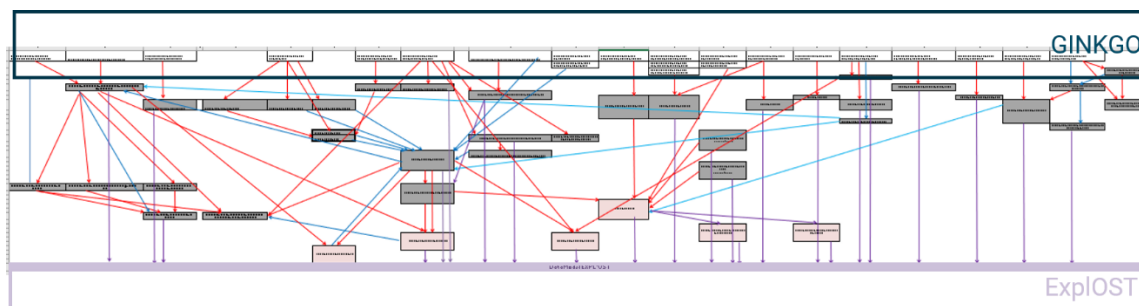


Figure 22 - Principe de la création du modèle de données ExplOST

L'impact sur les applications Qlikview est donc important. Cependant, le modèle orienté utilisateur ne propose pas de supprimer des données exploitées mais de les réorganiser. Il doit donc être possible de modifier le processus de mise à jour des données ExplOST en se basant sur le nouveau schéma.

Impact sur l'interface IPERU de rapprochement institutionnel

Cet impact n'a pas été mesuré (hors du périmètre de la mission).

Cette mesure apparaît nécessaire afin d'estimer la charge de travail à produire pour maintenir le fonctionnement de l'interface de rapprochement institutionnel. Cependant, comme dans le cas des applications Qlikview, le modèle orienté utilisateur ne proposant pas de supprimer des données exploitées mais de les réorganiser, il doit être possible de modifier le processus de mise à jour des données sous-jacentes à l'interface de rapprochement institutionnel.

Conclusion sur le scénario évolutif

La mise en place d'un tel scénario constituerait un investissement de la part du Hcéres.

Plusieurs avantages majeurs sont attendus de cet investissement.

- Dans la mesure où cette évolution se base sur les XML files dès le début, elle permet de fait de prendre en compte les différences entre les formats tagged et XML files.
- Le système GINKGO est vieillissant et l'actualisation des données est très coûteuse, une telle évolution permettrait de diminuer les coûts de fonctionnement, humains ou matériels (espace disque notamment), tout en améliorant la qualité des données produites

6. CONCLUSION GENERALE

Les deux scénarii analysés présentent chacun ses risques et ses avantages. Nous les résumerons comme suit :

- Le scénario conservateur est sans doute le moins coûteux à mettre en œuvre, même si une estimation précise est difficile, mais il conduit à continuer de supporter des délais d'actualisation et des coûts de maintenance élevés pour un résultat dont la qualité n'est pas satisfaisante.
- L'évolution du système constitue un investissement qui représente une amélioration des performances et une réduction des coûts de maintenance, et peut permettre au Hcéres de se concentrer sur des activités à plus forte valeur ajoutée

Comme il était pressenti par l'OST, le passage au format XML pour intégrer les données représente une opportunité d'améliorer le système d'information.

Enfin, au-delà d'un projet d'évolution de GINKGO, certains aspects du système d'information de l'OST n'étaient pas couverts par cette étude et il serait pertinent de les analyser. D'une part ils constituent des facteurs de complexité pour un projet d'évolution de GINKGO. D'autre part leur pertinence et leur adaptation aux besoins des utilisateurs doivent être réexaminés. Parmi ces aspects on pourra citer :

- Le calcul des indicateurs et la production des TSS
- L'usage de QlikView
- Le rapprochement institutionnel effectué dans le cadre d'IPERU.