Flux d’intégration

Giuseppe Berio

2023

Objectif d’un flux d’intégration

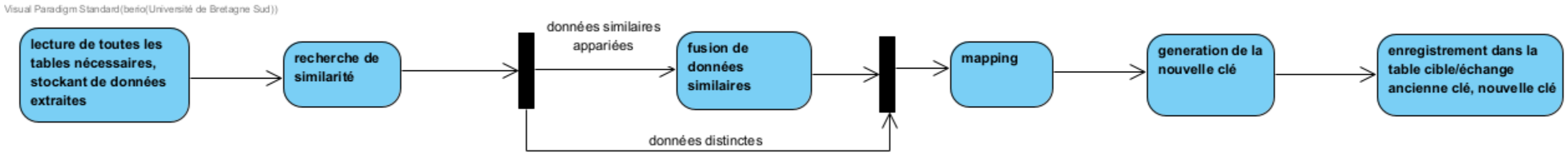
- Pour des tables du schéma intégré (SI) correspondantes à des superpositions de plusieurs éléments provenant de schémas distincts,
  - trouver les données correspondantes à un même objet, personne, phénomène, stockées sous ces plusieurs éléments issus de ces schémas distincts,
  - et les fusionner, pour ne pas créer des redondances dans les tables du SI
- Créer les données issues de la fusion, dans les tables du SI

Choix et contraintes

- Les **données en entrée** d'un flux d'intégration sont les **données extraites des sources et éventuellement transformées**
- Ces données extraites et transformées sont supposées être stockées sous un **format relationnel** (pour simplifier la discussion) mais cela n'est pas un vrai enjeu étant donné que les ETL traitent d'une manière uniforme plusieurs formats de stockage
- De la même manière, il est supposé de stocker les données en sortie du flux d'intégration (les **données intégrées**) sous un **format relationnel** ; il est également supposé d'utiliser des **clés de remplacement** (surrogate keys) pour ces données
- Il sera donc nécessaire **d'ordonnancer le flux d'intégration**, comme dans toute création de données dans une base de données relationnel :
  1) Commençant à créer les données dans les tables cibles du SI n'étant pas obtenus par superposition et ne contenant pas de clé étrangère
  2) Ensuite, créant les données dans les tables cibles ne contenant pas de clé étrangère et étant obtenues par superposition
  3) Ensuite, créant les données dans les tables cibles dépendantes (à savoir contenant des clés étrangères) des tables cibles traitées au point (2) précèdent
  4) Et ainsi suite
- Cet type d'ordonnancement suppose que pour les données à intégrer ne sont dépendantes que des données directement liées par une clé étrangère ; cela peut ne pas être une solution correcte à l'intégration de données

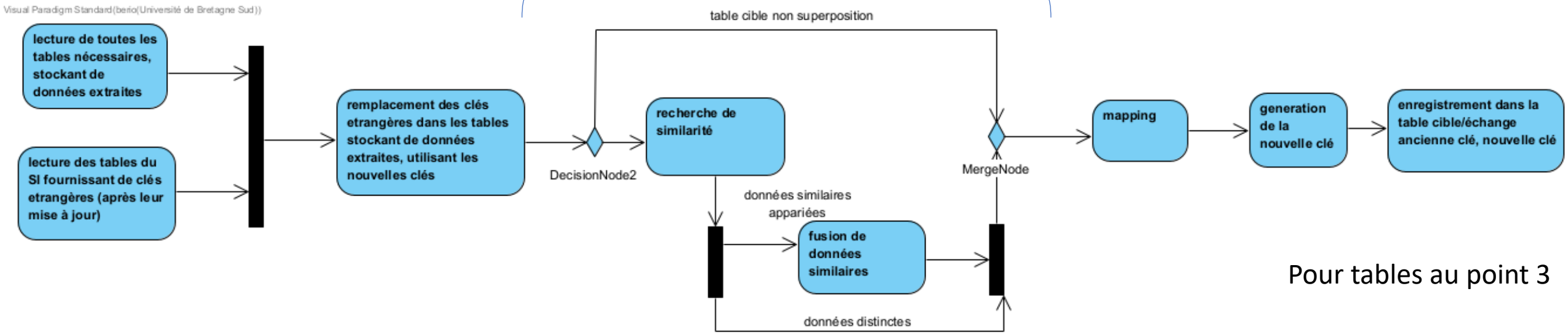
Flux d'intégration

Pour tables au point 1



Pour tables au point 2

dédoublonnage



Pour tables au point 3

Détails des activités

- Les activités de recherche de similarité et de fusion de données sont proches des celles à mettre en place lors de l'extraction pour le dédoublonnage
  - Cependant, la difficulté est généralement plus importante car les données proviennent de sources distinctes et donc toute fonction de similarité risque de générer **un plus grand nombre de faux positifs**
- Puisque de données peuvent être intégrées, il est nécessaire de générer des nouvelles clés, supposé être de clés de remplacement (surrogate keys), à chaque lancement des flux ; il est donc nécessaire prévoir un **stockage des anciennes clés** pour pouvoir faire aisément le remplacement de clés étrangères
  - Cependant, cela peut introduire des erreurs car il est toute à fait possible que les données similaires (donc fusionnées ou intégrées) ne soient pas toujours les mêmes entre 2 lancement de flux
- **Le mapping permet de créer les données intégrées dans la table cible du SI** ; cette activité peut se réaliser par une injection à partir d'une table, par union (ou par des « insert » multiples) ou par des opérations plus complexes

Focus sur la recherche de similarité

- La recherche de similarité peut correspondre à une différente terminologie : « entity identification, record linkage, duplicate detection,… »
- La recherche de similarité peut être réalisée utilisant des **techniques simples** sur les données issues des tables lues :
    - par exemple, des **simples jointures**
  - Dans ce cas, la fusion de données similaires est aussi opérée par des techniques simples (voir transparent suivant) :
  - Par exemple, **choix statique ou dynamique** d'une donnée issue d'une des tables, **agrégation de plusieurs valeurs (sum, max, min, avg,…)** provenant des différentes tables
- Si la recherche de similarité (ou elle est impossible) demande des **techniques plus avancées** (utilisant par exemple des **mesures de similarité générales sur lignes complètes**)
    - il est donc possible d'injecter toute les données lues dans une seule table et ensuite opérer une recherche de similarité (par une opération type UNION ALL)
  - Dans ce cas, la **fusion de données peut être opérée suivant la recherche de similarité (exemple, un médoïde par cluster)** mais elle peut aussi être opérée ensuite utilisant les techniques simples mentionnées ci-dessous

Focus sur la recherche de similarité (techniques simples)

- Contexte :
  - T1(A,B,C,D)=T2(A',B',C',E) WCI A=A', WCP B=B', C=C' (correspondance)
  - 1 seule table T(A,B,C,D,E) dans le schéma intégrée
- Pour l'opération de recherche de similarité n'est pas obligatoire de suivre la correspondance, notamment le WCI (car il peut ne pas constituer un vrai identifiant)
- La(les) requête(s) suivante(s) montre(nt) comment, en principe, la recherche de similarité est réalisable (ainsi que la fusion)

Select A, B, *f1(C),f2(D),f3(E)* → *fusion de données*
From
(Select **S1.T1.A as A, S2.T1.B as B, *S1.T1.C* as C,S1.T1.D as D,NULL as E**
From S1.T1,S2.T2
Where **SIM(S1.T1.A, S2.T2.A', S1.T1.B, S2.T2.B')>seuil → recherche de similarité**
group by S1.T1.A, S1.T1.B
UNION → nécessaire à la fusion
S2.T1.E Select **S1.T1.A as A, S2.T2.B' as B, *S2.T2.C'* as C,NULL as D**, *S2.T2.E as E*
From S1.T1,S2.T2
Where **SIM(S1.T1.A, S2.T2.A', S1.T1.B, S2.T2.B')>seuil → recherche de similarité**

« Group by » S1.T1.A, S1.T1.B → le représentant des similaires

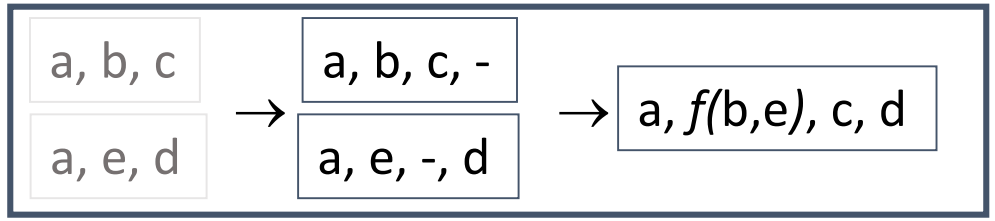
Select A, B, *f1(C),f2(D),f3(E)* → *fusion de données*
From
(Select **S1.T1.A as A, S2.T1.B as B, *f4(S1.T1.C, S2.T2.C')* as C,S1.T1.D as D, S2.T2.E as E**
From S1.T1,S2.T2
Where **SIM(S1.T1.A, S2.T2.A', S1.T1.B, S2.T2.B')>seuil → recherche de similarité**

« Group by » S1.T1.A, S1.T1.B → le représentant des similaires

Focus sur la recherche de similarité (techniques simples)

- Toutes les données exclues par la requête, sont de données propres à chaque table et donc devant être récupérées telles quelles et prise en compte pour le mapping
- « group by » en rouge n'est pas forcement la bonne syntaxe car dépendant de la forme de f1, f2, f3 (et de toute manière l'opération est effectuée par l'ETL utilisant un langage spécifique)
- Problème : ces requêtes ne garantissent pas que les données similaires forment une partition

Focus sur la fusion (simple) de données



Tuples présentant des valeurs (b,e) en conflit

a, b : attributs de jointure
e,b,c,d : autres attributs

Function	Description	Examples
Min, Max, Sum, Count, Avg	Standard aggregation	NumChildren, Salary, Height
Random	Random choice	Shoe size
Longest, Shortest	Longest/shortest value	First_name
Choose(source)	Value from a particular source	DoB (DMV), CEO (SEC)
ChooseDepending(val, col)	Value depends on value chosen in other column	city & zip, e-mail & employer
Vote	Majority decision	Rating
Coalesce	First non-null value	First_name
Group, Concat	Group or concatenate all values	Book_reviews
MostRecent	Most recent (up-to-date) value	Address
MostAbstract, MostSpecific, CommonAncestor	Use a taxonomy / ontology	Location
Escalate	Export conflicting values	gender
…	…	…

Focus sur le mapping

- Il y a 3 types de mapping qui peuvent être réalisés :
  - GAV (Global As View)
  - LAV (Local as View)
  - GLAV (Global Local as View)
- Pour un ETL standard et un schéma intégré, typiquement, GAV est celui utilisé correspondant à une « union » (ou union all)
- Mais le mapping GAV a des limites, notamment celle de ne pas pouvoir représenter des informations contextuelles relatives aux sources et non disponibles dans les sources (sauf enrichissement) ou celle de « bien fonctionner » lorsque le schéma intégré est parfait
- Les mapping GLAV/LAV permet un découplage entre les schémas de sources et le schéma cible

Flux de chargement (dans l’entrepôt)

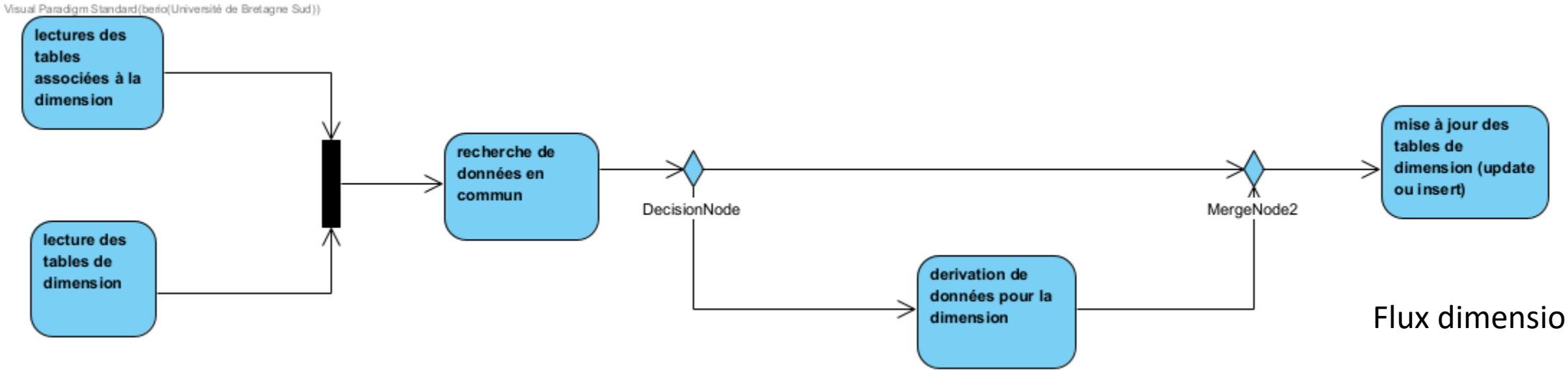
Giuseppe Berio

2023

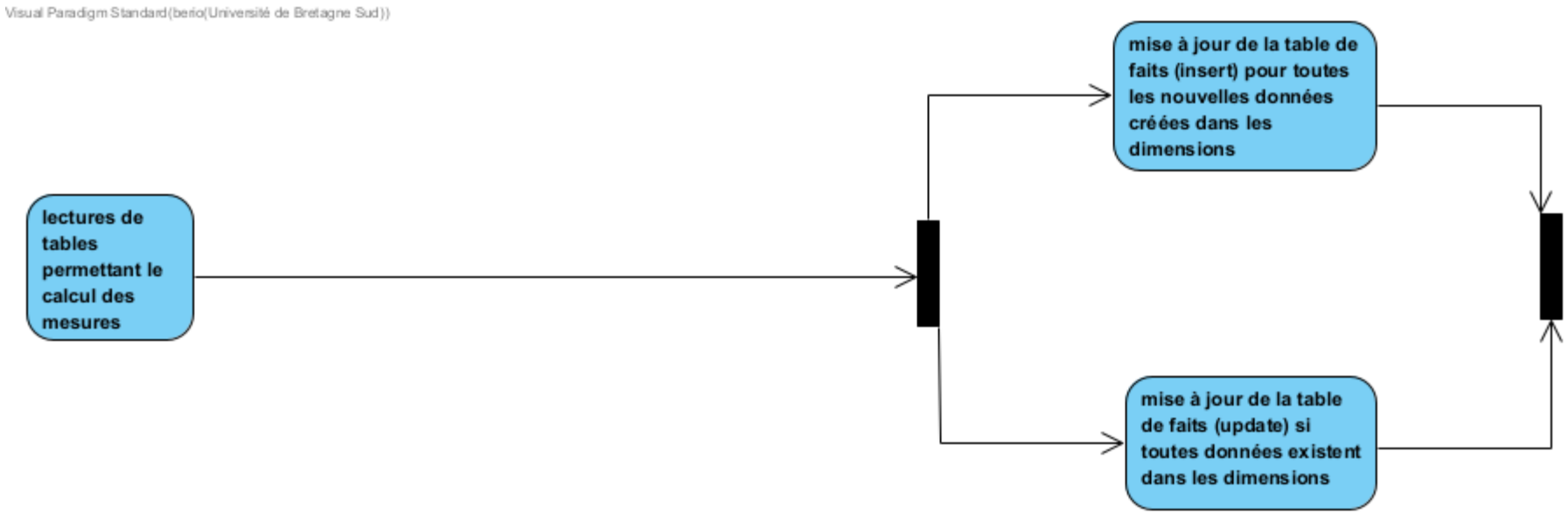
Principes

- Chargement de dimensions (pour les dimensions non préchargées)
  - Prise en compte de la modélisation SCD/RCD (nécessite un éventuel changement de clé et un calcul)
  - Prise en compte de l'existence de plusieurs tables de faits pour les dimensions conformes (schéma en constellation)
  - Difficultés :
    - les clés utilisées pour les données stockées sous le schéma intégré peuvent avoir été modifiées par 2 flux d'intégration successifs
    - Toute ancienne donnée n'existant plus dans les sources ne doit pas être supprimée dans les tables de dimension (les données sont ou bien mises à jour ou bien créées) car nécessaire pour « positionner » les faits passés ; il s'agit donc de rajouter les nouvelles données et de mettre à jour les données, potentiellement modifiées et encore disponibles dans les sources
- Chargement de la table de faits (pour une schéma flocon ou étoile)
  - Calcul des mesures (souvent correspondant a un « group by SQL »)
  - Les anciennes données n'existant plus dans les sources ne doivent pas être supprimées car elles permettent de « positionner » par rapport aux dimensions les faits passés
- Prise en compte des chargements successifs
  - Pour la table de faits, il s'agit de la mise à jour du calcul des mesures plutôt que le recalcul
  - Pour les dimensions, il s'agit principalement de repérer les données encore disponibles dans les sources et ayant subi une modification

Flux de chargement



Flux dimension



Flux table de faits

Dimensions préchargées

- Il s’agit de dimensions dont les données sont considérées indépendantes des sources
- L’exemple le plus typique est celui de la dimension « temps » constituées principalement par des dates, informations vacances, type d’année, numérotation semaine, décomposition en semestre/trimestre ou toute autre période d’intérêt pour l’entreprise ou organisation
- Exemple de script ORACLE pour créer des dates sur plusieurs années (10 ans environs) :

```
drop sequence s;
create sequence s start with 1 increment by 1;
insert into dimtemps select s.nextval as id, to_char((to_date(‘2023-12-31', 'YYYY-MM-DD')-level+1), 'dd’) as “day”, to_char((to_date('2018-12-31', 'YYYY-MM-DD')-level+1), 'mm’) as “month”, to_char((to_date('2018-12-31', 'YYYY-MM-DD')-level+1), 'YYYY’) as “year”, to_date(‘2023-12-31', 'YYYY-MM-DD')-level+1 as “date” FROM dual connect BY level <= 3650;
```