British Journal of Multidisciplinary and Advanced Studies 5(5),19-28, 2024
Engineering and Technology
Print ISSN: 2517-276X
Online ISSN: 2517-2778
https://bjmas.org/index.php/bjmas/index
Published by the European Centre for Research Training and Development UK

CONCEPTION D'UN DATA WAREHOUSE SUIVANT LA MODELISATION DE PROCESSUS ETL

Maxime Kasonga Badibanga

doi: https://doi.org/10.37745/bjmas.2022.04170 Published September 05, 2024

Citation :Badibanga M.K. (2024) CONCEPTION D'UN DATA WAREHOUSE SUIVANT LA MODELISATION DE PROCESSUS ETL, *British Journal of Multidisciplinary and Advanced Studies*, 5(5),19-28, 2024

Résumé. *Un processus ETL (Extract-Transform-Load) est très complexe en termes de flux de données et des tâches chargées de nettoyer, filtrer, normaliser et charger les données dans l'entrepôt de données. L'extraction des données à partir des sources, transformation permettant de livrer des données de qualité ayant une valeur pour l'analyse) chargement des données préparées dans l'entrepôt. Nous proposons dans cet article une modélisation des processus ETL au niveau conceptuel et logique, les modèles obtenus sont stockés sous forme de documents XML. Nous nous sommes basés sur l'approche de Panos Vassiliadis et al, tout en adaptant le métamodèle conceptuel et proposant un métamodèle au niveau logique*[1].

Mots clés : ETL ; XML, data warehouse

INTRODUCTION

Face à un marché très concurrentiel, les entreprises doivent disposer de grandes capacités d'analyse. L'information dont dispose l'entreprise constitue une ressource précieuse pour analyser, comprendre et agir en conséquence. Seulement, la grande difficulté est au niveau des quantités de données volumineuses stockées sous divers formats et modèles. En plus de cette hétérogénéité, ces données sources n'ont pas été constituées avec des perspectives d'analyse. Malgré les difficultés qu'elles présentent, ces données sont d'une valeur inestimable pour des applications d'analyse et d'aide à la décision. Avant d'être chargées dans un entrepôt, les données sources passent par un processus d'extraction, de transformation et de chargement, connu sous le nom ETL afin de les préparer et leur donner les propriétés nécessaires en termes de pertinence.

Vu la complexité et l'importance de l'ETL dans un projet décisionnel, nous nous intéressons dans cet article à la modélisation d'un processus ETL aux niveaux conceptuel et logique permettant d'anticiper la complexité et les aléas afin d'avoir de la visibilité sur tout le processus avant son implémentation. Dans cette problématique, Panos Vassiliadis et *al*. ont proposés un

[1] Vassiliadis, P., Simitsis, A., & Skiadopoulos, S. (2002). Conceptual Modeling for ETL Processes.In Proc. ACM 5th International Workshop on Data Warehousing and OLAP (DOLAP 2002), McLean, VA, USA November 8, 2002.

formalisme graphique adhoc permettant de modéliser les données sous forme de concepts caractérisés par un ensemble d'attributs[2].

La puissance du formalisme de Vassiliadis et *al.* est dans la description détaillée des tâches de transformation. L'approche est décrite par un métamodèle qui assure une extensibilité. La contribution de Juan Trujilio & Sergio Lujan Mora (ER2003) est basée sur une extension UML avec des profils pour la modélisation multidimensionnelle[3].

Notre contribution se résume en un certain nombre de propositions et compléments du métamodèle de Vassiliadis au niveau conceptuel pour modéliser de manière plus précise la réalité d'un processus ETL, Proposition d'un métamodèle pour le niveau logique afin de visualiser à un niveau très élevé les différents éléments ainsi que leurs relations.

Le premier point de cet article présentera le processus ETL de manière générale ainsi que les problèmes qu'il pose dans le cadre d'un projet décisionnel. On montrera pourquoi modéliser le processus ETL et que faut-il modéliser. La deuxième partie sera consacrée à l'approche de Vassiliadis pour la modélisation ETL en présentant ses concepts de base, ses principes et son métamodèle. Nous résumons, dans la troisième partie, nous allons présenter notre contribution sous forme de propositions par rapport à l'approche de Vassiliadis et nous terminons cet article par une conclusion et des perspectives.

Processus ETL

Un processus ETL (Extract-Transform-Load) sert à extraire des données à partir de diverses sources hétérogènes à préparer et à charger dans l'entrepôt de données. La difficulté du processus ETL est dans la diversité des données et leur hétérogénéité[4].

La figure 1 décrit l'environnement d'un processus ETL : la couche inférieure présente la partie statique, i.e. les données sources, le Data Staging Area (*DSA*) où s'opèrent les tâches de transformation et l'entrepôt de données (*DW*). Dans la couche supérieure, sont représentées les trois phases ETL à savoir extraction (*Extract*), transformation (*Transform & Clean*) et chargement (*Load*).

Afin de maîtriser cette complexité, les concepteurs préfèrent mettre toute la lumière sur ce processus ETL avant d'aborder l'implémentation physique.La modélisation du processus ETL aux niveaux conceptuel et logique en est une des solutions qui contribuent à la maîtrise de la complexité et des aléas du processus ETL. La modélisation devra formaliser tous les éléments d'un processus ETL de manière à comprendre le circuit des données depuis les systèmes sources jusqu'à l'entrepôt. Le mappage des données à différents niveaux est un aspect très important pour comprendre le cheminement des données.

[2] Vassiliadis, P., Quix, C., Vassiliou, Y., & Jarke, M. (2001). Data Warehouse Process Management. Information Systems, 26, 3, pp. 205-236

[3] Trujillo, J., & Luján-Mora, S. (2003). A UML Based Approach for Modeling ETL Processes in Data Warehouses. In Proceedings of 22nd International Conference on Conceptual Modeling (ER 2003), pp. 307-320, Chicago, IL, USA, October 13-16, 2003

[4] Simitsis, A. (2005). Mapping conceptual to logical models for ETL processes. In Proceedings of ACM 8th International Workshop on Data Warehousing and OLAP (DOLAP 2005), pp.: 67-76 Bremen, Germany, November 4-5, 2005.

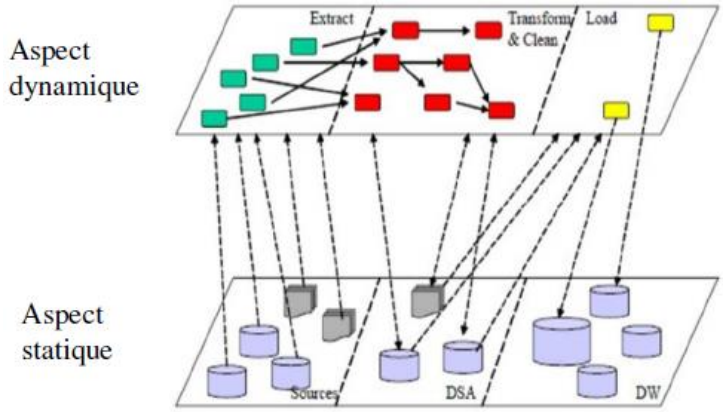


FIG.1.1. *Environnement d'un processus ETL. Panos Vassiliadis et al. (Dolap2002)*

L'approche de Vassiliadis

Il est intéressant de visualiser les aspects statiques (données) et dynamiques (tâches ETL) aux niveaux conceptuel, logique et physique. Le niveau conceptuel consiste à donner une idée très générale sur l'environnement du projet décisionnel à savoir : les besoins des utilisateurs en termes d'analyses (mesures et dimensions), les sources disponibles, transformations à faire subir aux données.

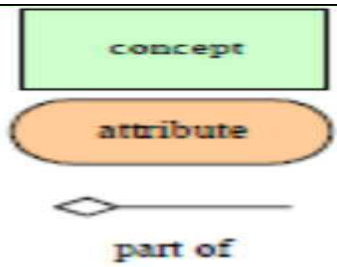
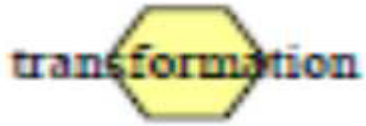
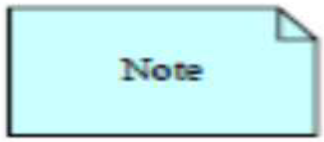
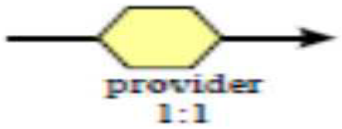
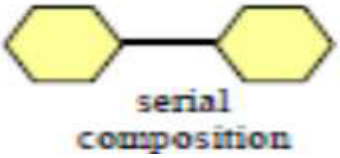
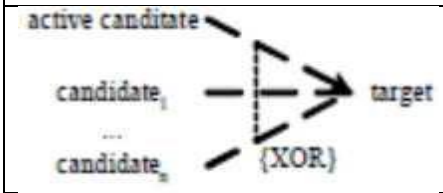
Le passage vers le niveau logique consiste à intégrer d'autres détails comme la séquence d'exécution des activités, différents schémas de données relatifs à une activité (inputs, outputs, paramètres, données rejetées). Le niveau physique qui décrit de manière complète le processus ETL doit préciser l'environnement sous lequel s'exécutera le processus : OS, nature des systèmes sources, infrastructure matérielle ainsi que les différents profils utilisateurs. Pour faire toute la lumière sur les tâches de transformations, l'approche représente l'activité à un niveau de granularité très fin : les attributs. Ces derniers sont valorisés et modélisés comme des entités à part entière et participent dans la représentation des détails des tâches de transformation.

Modélisation conceptuelle

Le formalisme proposé par Vassiliadis permet de représenter les différents objets manipulés dans le processus ETL ainsi que les associations reliant ces objets (mappage) vers ceux de l'entrepôt de données. La figure 3 présente un exemple de processus ETL relatif à la gestion d'un établissement universitaire. Les sources *marchandise (cursus en désignation)* et *scolarité (catégorie)* sont candidates pour l'entreposage. Actuellement, *c'est la première source qui* est retenue (*active candidate*).

Les données extraites à partir de cette source sont copiées dans le concept *S1.Client* pour opérer les transformations nécessaires avant leur chargement dans l'entrepôt (*DW.Effectif*). Le schéma montre comment sont traités les attributs du concept *S1.CLIENT* au niveau des transformations *SK* (*serrogate key*), *NN* (*not null*), *f1* (*fonction year()*), *agrégation γ* (*count()*). *PK* étant une contrainte (*Primary key*) sur les données *Année*, *désignation* et *Catégorie* de l'entrepôt.

British Journal of Multidisciplinary and Advanced Studies 5(5),19-28, 2024
Engineering and Technology
Print ISSN: 2517-276X
Online ISSN: 2517-2778
https://bjmas.org/index.php/bjmas/index
Published by the European Centre for Research Training and Development UK

SYMBOLES	DESIGNATION
	Représente une entité dans la source de données, dans le DSA ou dans l'ED Comme dans une modélisation E/R, ils permettent de définir les concepts Cette association permet de relier le concept à ses attributs
	Permet d'exprimer certaines contraintes sur le contenu de l'ED à travers les attributs de celui-ci (PK, FK, NOT NULL, …)
	Abstraction d'un bout ou d'un module complet de code exécutant une tâche ETL : (1) nettoyage/ filtrage de données (comme violation de la contrainte PK/FK, (2) transformations de données (comme une agrégation).
	Permet d'expliquer des choix de conception, de préciser une sémantique ou une contrainte à vérifier en temps réel (temps d'exécution, événements, erreurs, …)
	Représentent le mappage entre les données sources (input) et les données de l'ED (output) via une transformation. Le cas simple (1 :1) représente le cas où une donnée source (input) donne en sortie, à travers une transformation, à une donnée de l'ED (output).
	Le cas général (N:M) représente le cas où un ensemble de données sources (inputs) transformées donneront naissance à plusieurs données de l'ED (outputs)
	Permet de modéliser le cas où dans une association « Provider » les données passent par plusieurs transformations avant de donner naissance aux données de l'ED.
	Permet de préciser les sources de données candidates à l'alimentation de l'ED en mettant en relief la source active. Une des sources pourra être utilisée en même temps (XOR)

TAB. 1 – *Formalisme de Vassiliadis pour la modélisation au niveau conceptuel*

Pour que le formalisme soit générique, Vassiliadis et al. ont proposé un métamodèle (figure 4) regroupant l'ensemble des éléments, sous forme de métaclasses, pouvant intervenir dans un processus.

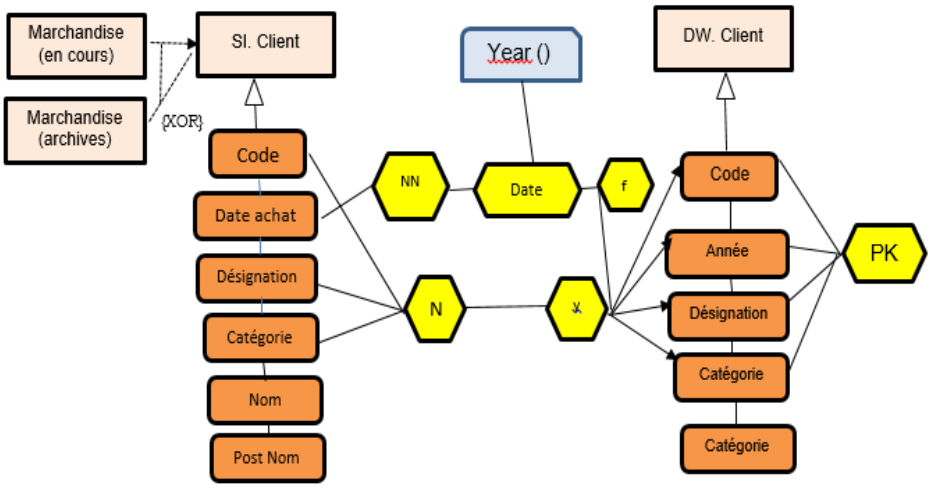


FIG. 3 – *Exemple d'un processus ETL relatif à un établissement de vente de marchandise*

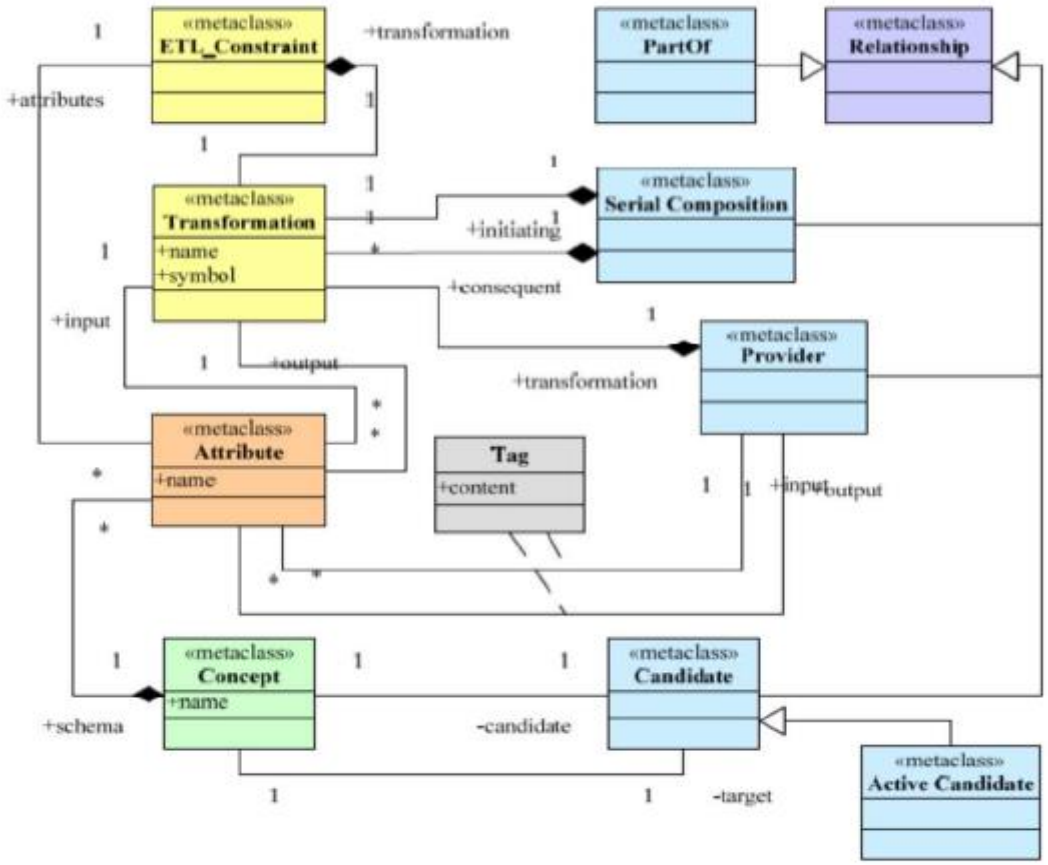


FIG. 4 – Métamodèle proposé pour la modélisation au niveau conceptuel

Modélisation logique

Le modèle logique, appelé aussi graphe d'architecture, capture les flux de données à partir des sources vers l'entrepôt de données grâce à une suite d'activités synchronisées chargées de préparer les enregistrements de données. Il s'agit de préciser le type des données utilisées lors des traitements, les inputs/outputs, paramètres de chaque activité, la séquence des activités.
Le graphe d'architecture comporte des sommets et des arêtes. On distingue plusieurs types de sommets :

a) Les entités élémentaires
- DataTypes: Chaque type de données T est caractérisé par un nom et un domaine,
- Attributes: Les attributs sont caractérisés par un nom et un type de données.
- Schema: est une liste finie d'attributs. Toute entité caractérisée par un ou plusieurs schémas est appelée entité structurée.

b) RecordSet: Un enregistrement est défini comme l'instanciation d'un schéma à une liste de valeurs appartenant aux domaines des attributs respectifs au schéma.

c) Function: Il est supposé l'existence d'un ensemble fini de types de fonctions. Un type de fonction comporte un nom, une liste finie de types de données des paramètres et un seul type de retour de données.

d) Activity: Les activités sont considérées comme des abstractions logiques représentant des parties ou des modules de codes complets. Elles sont représentées par un langage LDL qui, d'une part, définit le code source d'une activité et d'autre part évite le traitement des spécificités d'un langage particulier.

Une activité est formellement décrite par un nom (*Name*), schéma d'entrée (*Input Schemata*), schéma de sortie (*Output Schema*), schéma des rejets (*Rejections Schema*), liste des paramètres (*Parameter List*), sémantique opérationnelle des sorties (*Output Operational Semantics*), sémantique opérationnelle des rejets (*Rejection Operational Semantics*).
Les différents types de relations (arêtes) constituant un graphe d'architecture sont :

- *PartOf*: Met en relation les attributs et les paramètres avec les activités, enregistrements ou fonctions auxquels ils appartiennent.
- *Instance-Of*: Permet de capturer les informations sur le typage des attributs et des fonctions.
- *Regulator*: Cette relation est définie entre les paramètres d'une activité et les termes (attributs ou constantes) qui alimentent cette activité.
- *Provider*: Cette relation capture le passage des données entre fournisseurs (*Providers*) et consommateurs (*Consumers*) par la relation Provider entre les attributs des schémas concernés.
- *Derived provider*: Cas particulier de la relation Provider. Cette relation est utilisée lorsque les attributs de sortie sont générés par la composition des attributs d'entrée et des paramètres de l'activité.

La figure 5 présente un schéma du processus au niveau logique de l'exemple d'un établissement de vente de marchandises :

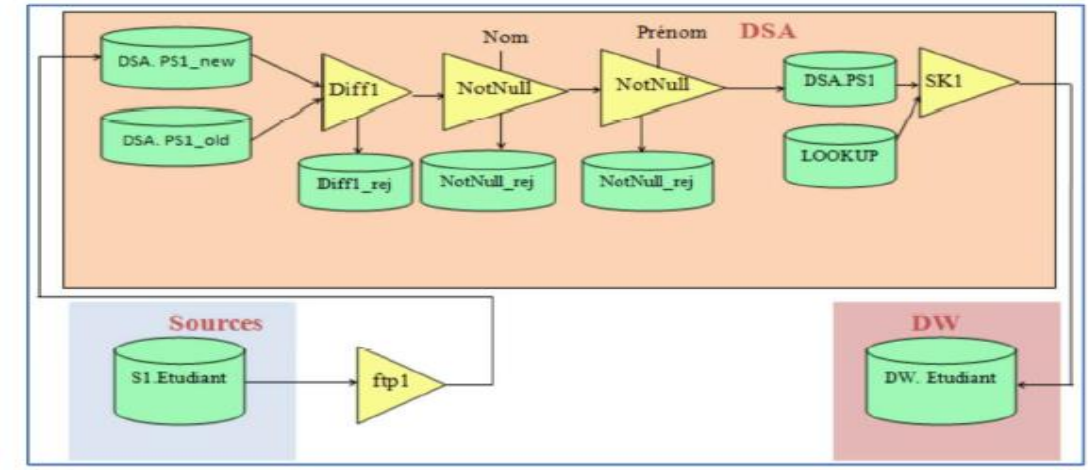


FIG. 5 – *Exemple du processus ETL au niveau logique relatif à un établissement Universitaire*

Contribution

Une analyse de l'approche de Vassiliadis nous a permis de comprendre le lien entre les différentes parties, de mettre en valeur certains aspects implicites et d'apporter quelques améliorations. Au niveau conceptuel, nous avons proposé quelques modifications au niveau du métamodèle dans le but de le rendre plus fin et plus exhaustif. Pour le niveau logique, nous avons proposé un métamodèle qui décrit, de manière similaire à celui du niveau conceptuel, les éléments manipulés au niveau d'un modèle logique. Voici un résumé de notre contribution :

1) En analysant l'environnement d'un processus ETL (figure 1) et les notations proposées pour la modélisation d'un processus ETL (tableau 1), nous avons découvert le lien non explicité dans les papiers de Vassiliadis. Nous pourrons déduire et délimiter les différentes phases (sources de données, extraction, DSA, transformation, entrepôt de données) à partir du schéma d'un processus comme le montre l'exemple de la figure 6.
2) Dans le métamodèle de Vassiliadis, une partie seulement des schémas sources, DSA et entrepôts de données est représentée, i.e les concepts de données nécessaires aux besoins d'analyse dépourvues des relations entre eux. En rajoutant une association réflexive au niveau de la métaclasse « *Concept* » avec « *Attribut* » comme metaclasse-associative pour représenter l'attribut assurant le lien (*Foreign Key*), on aura représenté tout le contenu des schémas des données. Pour l'entrepôt en particulier, on aura le schéma en étoile

British Journal of Multidisciplinary and Advanced Studies 5(5),19-28, 2024
Engineering and Technology
Print ISSN: 2517-276X
Online ISSN: 2517-2778
https://bjmas.org/index.php/bjmas/index
Published by the European Centre for Research Training and Development UK

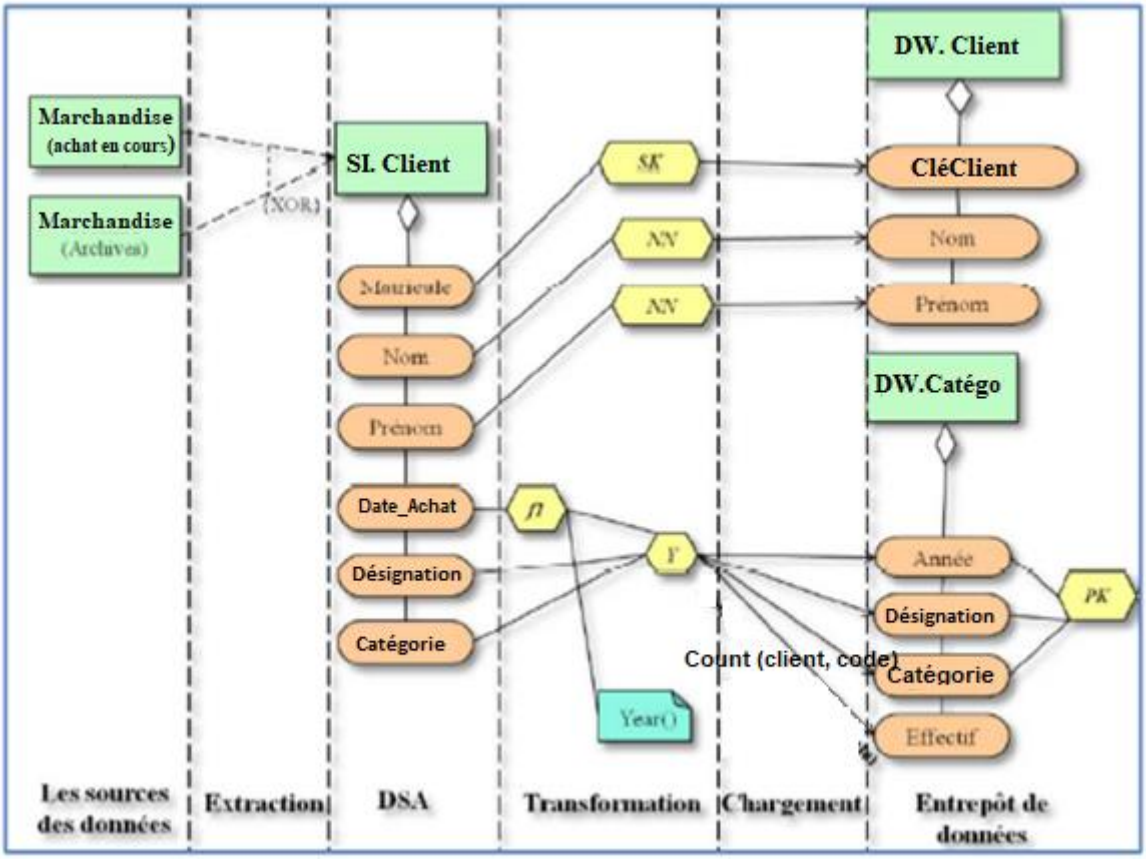


FIG. 6 – *Délimitation des différentes étapes dans le schéma du processus*

3) Dans le métamodèle, la Cardinalité (*1:1)* de la métaclasse « *candidate* » avec « concept » ne représente pas de manière naturelle la réalité de la relation. Une relation « *candidate* » possède plusieurs candidats sources et un seul candidat cible. Pour exprimer ceci, nous proposons alors de mettre une cardinalité (*1..**) dans l'association *candidate* au lieu de '1'. Si on note *R* la relation candidate, *C1, C2, C3* les concepts candidats et *C4* comme concept cible, les instances des deux associations *Candidate* et *Target* se présentent comme suit :

Relation	*Concept candidat*
R	*C1*
R	*C2*
R	*C3*

Relation	*Concept cible*
R	*C4*

TAB. 3 – Instances des relations *Candidate* et *Target* avec une cardinalité '1'

4) Pour lui donner plus de sens dans le métamodèle, la relation "*Part-Of*" représentée comme métaclasse doit être associée avec la métaclasse "*Concept*" et la métaclasse "*Attribut*" pour représenter que tel attribut est un constituant de tel concept.

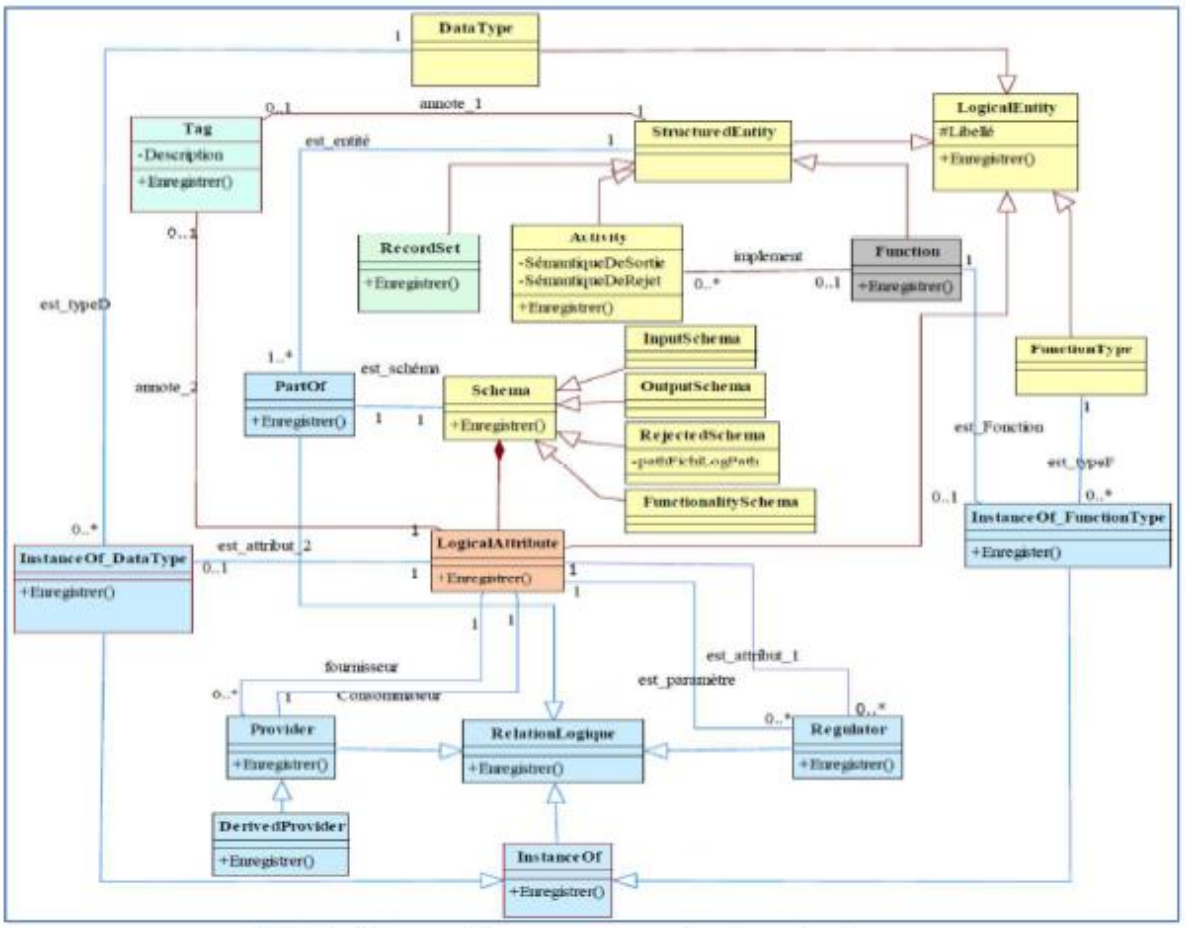


FIG. 8 – *Métamodèle proposé pour le niveau logique*

5) La métaclasse "*Tag*" telle que conçue ne permet pas d'utiliser une note partout dans le modèle conceptuel. Il serait plus intéressant de la relier à la métaclasse "*Relation*" (afin d'impliquer toutes les relations), les attributs, les concepts et les transformations.

6) Pour le niveau logique, nous avons proposé un métamodèle (voir figure 8)

Conclusion et perspectives

L'article a mis en valeur l'importance du processus ETL dans le cadre d'un projet décisionnel. La modélisation de ce processus est un moyen qui facilite la compréhension des détails de fonctionnement de l'ETL et permet alors de maîtriser sa complexité et anticiper sur les éventuels problèmes et risques avant l'implémentation ou le paramétrage de l'outil ETL.

L'approche de Vassiliadis étant une des plus intéressantes dans ce domaine. L'étude de cette approche nous a permis de découvrir des aspects très intéressants dans les processus ETL.

Notre contribution au niveau de l'approche se résume en un ensemble de remarques et de propositions pour affiner davantage le métamodèle et refléter au mieux la réalité d'un processus ETL.

Le niveau physique, qui modélise le processus ETL dans un environnement caractérisé par les moyens techniques (équipements et environnement logiciel) et les profils utilisateurs

nécessaires pour le bon fonctionnement du processus, mérite aussi une bonne analyse, proposition d'un formalisme, d'un métamodèle. Le mappage Conceptuel-Logique et Logique-physique est un aspect très intéressant qui mérite du travail en profondeur afin d'assurer la génération des modèles logiques et physique de manière semi-automatique mais avec un moindre effort pour le concepteur.

Références Bibliographique

1. Vassiliadis, P., Simitsis, A., & Skiadopoulos, S., *Conceptual Modeling for ETL Processes.In Proc. ACM 5th International Workshop on Data Warehousing and OLAP*, McLean, VA, USA November 8, 2002.
2. Vassiliadis, P., Quix, C., Vassiliou, Y., & Jarke, M., *Data Warehouse Process Management. Information Systems*, 26, 3, pp. 205-236, 2001.
3. Trujillo, J., & Luján-Mora, S. (2003). *A UML Based Approach for Modeling ETL Processes in Data Warehouses. In Proceedings of 22nd International Conference on Conceptual Modeling* (ER 2003), pp. 307-320, Chicago, IL, USA, October 13-16, 2003
4. Simitsis, A., *Mapping conceptual to logical models for ETL processes. In Proceedings of ACM 8th International Workshop on Data Warehousing and OLAP*, pp.: 67-76 Bremen, Germany, November 4-5, 2005.