

BI2 : Un profil UML pour les Indicateurs Décisionnels

Sandro Bimonte*

*Irstea, TSCF, 9 Av. Blaise Pascal, 63178, Aubière, France
sandro.bimonte@irstea.fr

Résumé. Aujourd'hui de plus en plus de données sont disponibles pour une analyse décisionnelle et reposent sur des indicateurs décisionnels. Bien que différentes technologies décisionnelles aient été développées, nous constatons le manque d'un cadre conceptuel pour la définition et l'implémentation de ces indicateurs. Dans ce papier, nous présentons une première classification de ces indicateurs. De plus, motivés par le besoin d'un formalisme pour la définition de ces indicateurs à un niveau conceptuel, nous présentons un profil UML BI2 qui permet de représenter des indicateurs OLAP, OLTP et stream. Nous présentons également leur implémentation dans les outils industriels existants.

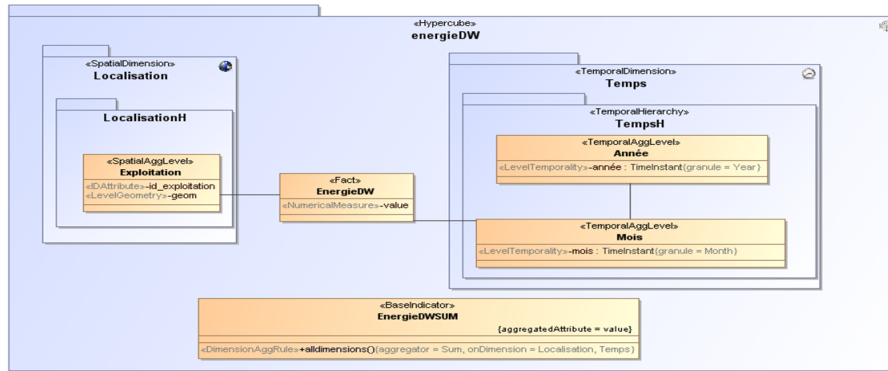
1 Introduction

Ces dernières années, des nouvelles technologies d'acquisition de données ont été développées permettant la collecte d'énormes quantités d'informations (*Big Data*). Il s'agit de données variées (données spatiales, multimédia, etc.). Ces données sont issues des réseaux sociaux, des réseaux de capteurs (mobiles), des satellites, etc. Sous ce déluge de données, plusieurs méthodes pour le stockage et l'interrogation de ces données ont été proposées (Cuzzocrea et al., 2013). De nouvelles possibilités d'analyse sont offertes aux décideurs qui disposent d'une variété importante de données et d'outils d'aide à la décision (Chen et al., 2012). Les technologies d'aide à la décision, connues sous le terme de Business Intelligence (BI), intègrent les Entrepôts de Données (EDs) et l'OLAP, le reporting, le Data Mining, les flots de données (i.e. stream data), etc. En conséquence, la mise en place d'un système décisionnel n'est plus aujourd'hui confinée à une seule catégorie d'indicateurs décisionnels. Les décideurs peuvent s'appuyer sur plusieurs méthodes et donc plusieurs technologies de BI. D'autre part, la modélisation conceptuelle des systèmes complexes (systèmes d'information, EDs, etc.) est reconnue comme un élément indispensable à la réussite des projets de BI (Turlone, 2003). Cette modélisation permet aux concepteurs de définir des modèles faciles à comprendre par les décideurs. Dans ce contexte, UML est considéré comme le standard orienté-objet pour la modélisation des différents aspects des systèmes logiciels. En effet, UML offre un outil puissant de formalisation aux concepteurs et décideurs, lors des phases d'élaboration et de mise en oeuvre des schémas de données. Il peut être également interprété par des ateliers de génie logiciel (AGL). Dans ce travail nous nous intéressons aux indicateurs décisionnels sous la forme de requêtes d'agrégation (par exemple la moyenne des ventes par mois et par région). Ce type de requêtes est essentiellement géré via les EDs et les systèmes OLAP, car les données sont organisées selon

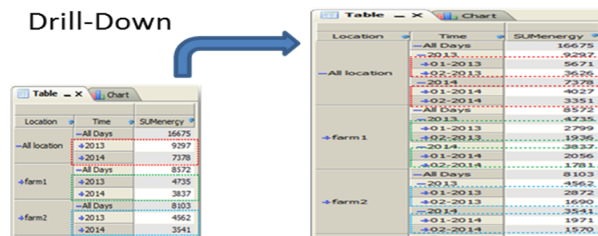
le modèle multidimensionnel qui définit les concepts de dimensions (les axes d'analyse) (par exemple temps et localisation) et les faits (les sujets d'analyse) (par exemple les ventes). Les données entreposées sont donc agrégées le long des hiérarchies des dimensions (par exemple jour, mois, année) avec des fonctions d'agrégation (i.e. min, max, sum, etc.). Différents formalismes conceptuels ont été proposés pour les EDs (Boulil et al., 2015). Par contre les EDs ne représentent pas la seule solution technique possible pour calculer ce type d'indicateurs à partir d'une base de données. Souvent une navigation hiérarchique et/ou de données historiques n'est pas nécessaire - des requêtes continues sur des flots de données peuvent par exemple être utilisées. En s'appuyant sur un cas d'étude relatif aux consommations énergétiques agricoles (Bimonte et al., 2014a), nous proposons dans ce travail une première classification de ces types d'indicateurs, ainsi que l'utilisation de technologies existantes pour leur implémentation (Sec 4). Motivés par le manque de cadre conceptuel pour ces indicateurs (Sec. 2), nous proposons un formalisme conceptuel unique pour leur représentation (Sec 5) basé sur un profil UML (BI2 UML Profile). BI2 UML Profile, implémenté dans l'outil AGL commercial Magic-Draw, permet de représenter les différents types d'indicateurs décisionnels dans un seul outil de conception. Par conséquent, ceci facilite la conception (l'échange avec les décideurs avant la phase d'implémentation) de ces nombreux indicateurs, car le même formalisme (UML) est utilisé, ce qui permet d'aboutir à la définition d'indicateurs décisionnels qui répondent aux besoins fonctionnels des décideurs. En effet, une des raisons de l'échec des projets BI est souvent l'inadéquation entre les solutions BI mises en place et les demandes des décideurs (Bimonte et al., 2014b). De plus, la conceptualisation de ces différents indicateurs devient nécessaire vu que les technologies utilisées pour leur implémentation sont très différentes et complexes. Par conséquent une "bonne" phase de conception permet une mise en place correcte des architectures nécessaires (OLAP, DSMS, etc.) et de leur implémentation, ce qui se traduit par des gains en terme de temps ingénieurs (Bimonte et al., 2014b).

2 Préliminaires : ICSOLAP UML Profil

Les profils UML constituent un moyen pour étendre et adapter UML à des domaines ou des plateformes particulières en étendant ses méta-classes (classe, propriété, etc.). Un profil est défini en utilisant trois mécanismes d'extension : les stéréotypes, les tagged values et les contraintes. Un stéréotype est une extension d'une méta-classe UML, et est représenté en utilisant la notation « nom » et / ou un icône. Les tagged values sont des méta-attributs ; ils sont définis comme des propriétés de stéréotypes. Enfin, un ensemble de contraintes doit être attaché à chaque stéréotype afin de définir leur sémantique et leurs contraintes d'utilisation lors de la conception de modèles. Ces contraintes sont souvent définies en utilisant OCL (Object Constraint Language). Les contraintes OCL peuvent être appliquées au niveau méta-modèle (par exemple, le profil UML) et au niveau du modèle (l'instance du profil UML). Les avantages de l'utilisation d'UML pour la modélisation des entrepôts de données complexes ont déjà été soulignées dans plusieurs travaux BI (Mazon et Trujillo, 2008). Dans cette section, nous décrivons le profil proposé par (Boulil et al., 2015) qui est utilisé dans notre profil BI2. (Boulil et al., 2015) proposent un profil UML pour la modélisation des entrepôts de données spatiales qui permet de représenter conceptuellement tous les aspects avancés de la modélisation spatio-multidimensionnelle. Le profil présente des stéréotypes pour chaque élément spatio-multidimensionnel, comme par exemple : «Fact» pour le fait, «SpatialAggLevel» pour



(a)



(b)

FIG. 1 – *EnergieDW* : a) modèle multidimensionnel, b) visualisation requête OLAP (Q2)

les niveaux des dimensions spatiales, etc. Un fait est composé de mesures «Measure». Un «AggLevel» est composé d'attributs «Dimensional attribute» et peut être thématique, spatial ou temporel. Les faits sont inclus dans un package «Hypercube». Dans ce profil, pour les règles d'agrégation, les auteurs ont défini un stéréotype «BaseIndicator» représentant un ensemble de fonctions d'agrégation appliquées à une mesure. Un exemple est donné en Figure 1. (Bouilil et al., 2015) définissent aussi une implémentation automatique du modèle spatio-multidimensionnel en Oracle Spatial et le serveur OLAP Mondrian.

3 Motivation

Dans cette section, nous présentons un cas d'étude portant sur les consommations énergétiques des exploitations agricoles. Ce cas d'étude est issu d'un projet appelé EDEN financé par le Ministère de l'Agriculture (Bimonte et al., 2014a). En particulier, il s'agit de contrôler les consommations électriques en production laitière. Pour cela, comme montré en Figure 1a, un ED Spatial a été mis en place en utilisant le profil UML ICSOLAP décrit en Section 2. Cet ED permet l'analyse des données historiques (sur plusieurs mois) des quantités totales d'électricité. Avec ce modèle multidimensionnel, les décideurs peuvent par exemple *Visualiser la consommation totale par mois et année, et pour toute l'exploitation* (Requête Q2 - Tableau 1). La visualisation de cette requête est présentée en utilisant le client OLAP JRubik. L'utilisateur

peut passer du niveau année au niveau mois avec une simple opération de Drill-Down sur la table de pivot (Figure 1b). Les affichages graphiques \acute{n} par année \acute{z} et \acute{n} par mois \acute{z} sont donc synchronisés dans le client OLAP, selon le paradigme d’analyse visuelle des \acute{n} vues synchronisées \acute{z} (Andrienko et al., 2003). Les sources des données sont issues des capteurs installés sur les agroéquipements des exploitations agricoles, et relevées à chaque seconde. Les exploitants souhaitent utiliser ces données dans deux scénarios différents : (1) *Alerte* et (2) *Analyse*. Dans le premier cas (Middelfart et Pedersen, 2010), les décideurs sont intéressés à être alertés dans le cas où une machine présente un dysfonctionnement ou un mauvais réglage (consommation trop importante par exemple). Une requête possible pour cette tâche peut être la requête Q1 du tableau 1. Dans le cas de l’Analyse, les exploitants souhaitent analyser les consommations en utilisant le modèle multidimensionnel de Figure 1a, mais aussi par exemple analyser la consommation énergétique par jour par capteur (Q3- Tableau 1) sans avoir besoin d’accéder à d’autres affichages graphiques à partir de celui par jour et capteur.

Requête	
Q1	Toutes les 3 secondes : Quelle est la consommation totale, maximale et minimale d’électricité au cours des 3 dernières secondes ?
Q2	Quelle est la consommation totale d’électricité par exploitation, mois et année ?
Q3	Quelle est la consommation totale d’électricité par jour et capteur ?

TAB. 1 – Requetes décisionnelles

4 BI2 UML profil

4.1 Cadre conceptuel

Dans cette section nous décrivons les propriétés de différents types de requêtes que nous ciblons, afin d’identifier des caractéristiques génériques utilisables dans la définition conceptuelle des indicateurs et dans leur implémentation. En première analyse, ces requêtes ont comme point commun l’agrégation d’une valeur numérique. Considérons maintenant la requête Q3. Cette requête porte sur des données historiques. Le décideur souhaite connaître la consommation énergétique pour tous les mois, depuis la mise en place de l’indicateur. Or, aucune fenêtre temporelle n’est définie. Nous appelons ce paramètre *Données Historiques*, il s’agit de données qui sont seulement ajoutées à la base de données sans qu’aucune donnée ne soit effacée. De la même façon, la requête Q2 porte aussi sur des données historiques. A l’inverse, la requête Q1 prend simplement en considération une fenêtre temporelle de données, en particulier les données des 3 dernières secondes. Dans les requêtes Q2 et Q3 le décideur ne spécifie pas à quel moment cette requête sera exécutée, il s’agit en effet d’une analyse ponctuelle sur les données, à la demande. A l’inverse, dans la requête Q1, le temps d’exécution de la requête est inclus dans la requête même (ex : toutes les 3 secondes), ce qui crée donc un lien entre les données et la fréquence d’exécution. Nous appelons ce paramètre *Fréquence continue*. Intéressons-nous maintenant aux requêtes Q2 et Q3. Ces requêtes, en apparence similaires, possèdent une différence importante : la possibilité de naviguer entre les indicateurs. Dans la requête Q2, il est expressément représenté le besoin de pouvoir lier entre eux les deux

affichages graphiques *par année et par mois*. Il s'agit donc de permettre une navigation hiérarchique dans l'outil de restitution, du niveau mois au niveau année et vice-versa (comme montré en Figure 1b). Nous appelons ce paramètre *Navigation hiérarchique*. Une fois ces critères établis, il est possible de définir, par croisement, des classes d'indicateurs et éventuellement leur associer une technologie décisionnelle, comme montré dans le Tableau 2.

1. Les indicateurs de type *Données Historiques, Fréquence non continue, Navigation hiérarchique* (**Indicateur OLAP**) correspondent aux indicateurs de type OLAP, comme décrit précédemment.
2. Les indicateurs de type *Données Historiques, Fréquence non continue, Navigation non hiérarchique* (**Indicateur OLTP**) correspondent aux indicateurs définis classiquement dans les outils de reporting comme SpagoBI, Jasper, etc., qui peuvent être implémentés à partir d'un système OLTP classique via des requêtes SQL.
3. Les indicateurs de type *Données non Historiques, Fréquence continue, Navigation non hiérarchique* (**Indicateur Stream**) font référence aux indicateurs calculés au sein des Data Stream Management Systems (DSMSs) (en anglais).

(Golab et Ozsu, 2010) définissent un flot de données (i.e. data stream) comme une séquence continue et ordonnée d'éléments en temps réel. Ainsi, il est impossible de contrôler l'ordre d'arrivée des éléments et de les stocker. Les requêtes sur les flots de données fonctionnent en continu sur une période de temps et retournent progressivement les résultats à fur et à mesure que de nouvelles données arrivent. Par conséquent, le stockage et l'analyse des flots de données nécessitent notamment des modèles de données, des langages et des outils d'interrogation spécifiques : les DSMSs (Data Stream System Management). La principale différence entre les SGBDs et les DSMSs est que dans les DSMSs il est possible d'interroger des données qui ne sont pas stockées de façon permanente. Dans un DSMS, comme les données ne sont pas permanentes, de nouveaux types de requêtes sont définis en utilisant le concept de fenêtre qui définit un intervalle temporel ou quantitatif sur les éléments du flot. Comme indiqué dans le tableau 2, toutes les classes d'indicateurs ne peuvent pas être implémentées dans les outils existants. En effet, pour des raisons de performances liées à la volumétrie des données, les requêtes continues portent sur des données non historiques et non hiérarchiques. Les technologies OLTP et OLAP ne disposent pas des mécanismes propres aux DSMSs pour la gestion de la continuité de l'interrogation. Il est important de souligner que pour les indicateurs de type *Données non Historiques, Fréquence continue, Navigation hiérarchique*, seul l'outil ActivePivot (QuartetFS, 2014) existe aujourd'hui dans le commerce. Dans le reste de cette section nous présentons notre profil UML, qui étend ICSOLAP, pour la définition des trois classes d'indicateurs décrits précédemment. Il est évident que les indicateurs de type *Données Historiques, Fréquence non continue, Navigation hiérarchique* peuvent être modélisés en utilisant le profil UML présenté en Sec 2.

4.2 Indicateur OLTP

Les indicateurs de type *Données Historiques, Fréquence non continue, Navigation non hiérarchique* sont modélisés en utilisant le profil décrit en Figure 2. Nous définissons un stéréotype de package «IndicatorsOLTP» qui contient un ensemble d'indicateurs («Indicator»). Une classe «Indicator» représente une requête de type *Données Historiques, Fréquence non*

Requête	Données		
Navigation	Fréquence	Historiques	Non Historiques
Hiérarchique	Continue		
Hiérarchique	Non Continue	OLAP (Q2)	OLAP
Non Hiérarchique	Continue		DSMS (Q1)
Non Hiérarchique	Non Continue	OLTP (Q3)	OLTP

TAB. 2 – Classes d'indicateurs et leur éventuelle implémentation

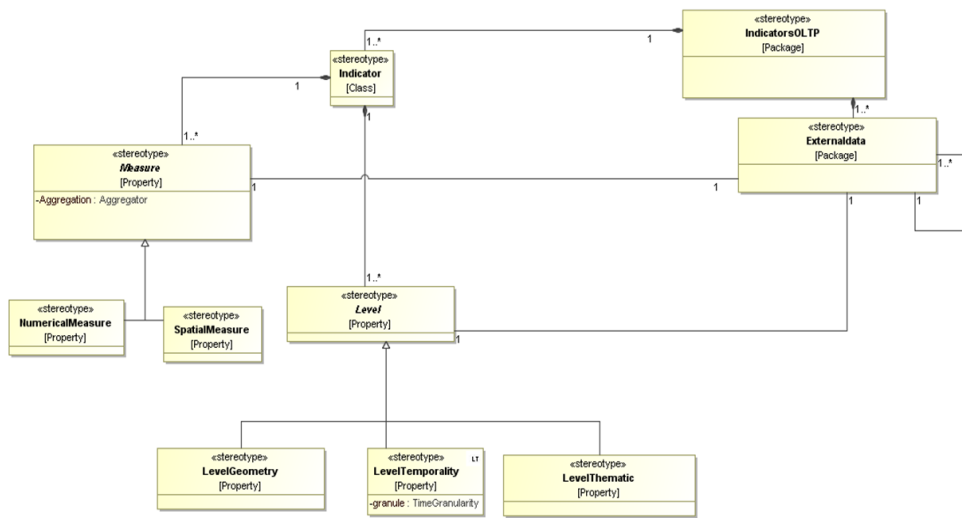


FIG. 2 – BI2 UML profil : IndicatorOLTP

continue, Navigation non hiérarchique. Elle contient un ensemble d'attributs de type «Mesure» et «Level». «Mesure» représente la valeur qui est agrégée, et peut être de type numérique («NumericalMeasure») et spatial («SpatialMeasure»). Une mesure est agrégée en utilisant une fonction d'agrégation qui est représentée par une tagged value («Aggregation») de «Measure». Un indicateur est aussi caractérisé par un ensemble de niveaux («Level») qui définissent la granularité de l'agrégation. Un niveau est représenté comme un attribut de «Indicator». Trois types différents de niveaux sont définis : temporel, spatial et thématique. Un exemple d'indicateur pour la requête Q3 est représenté en Figure 3, où la classe **SommeEnergie** contient deux niveaux (jour et capteur) et une mesure **value** avec une fonction d'agrégation SUM. Chaque attribut (mesure ou niveau) est associé à un package «Externaldata» qui représente l'ensemble de données sources qui sont utilisées pour définir cet attribut. Par exemple, dans la figure 3 la mesure **value** est issue de la classe **energie**. Un package «Externaldata» peut être associé à un autre package si une relation existe entre les classes qu'elles contiennent. Par exemple dans la figure 3, le package **energie** utilise les packages pour les données spatiales et temporelles.

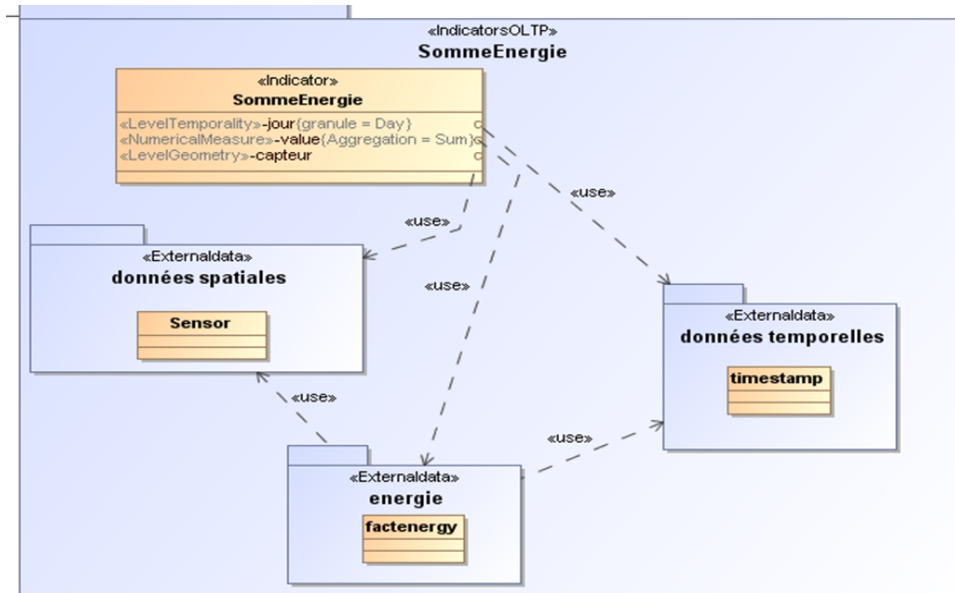


FIG. 3 – BI2 UML profil : IndicatorOLTP - Q3

4.3 Indicateur Stream

Un indicateur de type *Données non Historiques, Fréquence continue, Navigation non hiérarchique* est représenté par le stéréotype `<<IndicatorStream>>` qui étend `<<Indicator>>` (Figure 4). Il est composé par des niveaux d'agrégation qui peuvent être de type thématique `<<LevelThematic>>` et spatial `<<LevelGeometry>>`. Un `<<IndicatorStream>>` est aussi composé par un attribut qui représente la fenêtre d'agrégation glissante sur le flot de données `<<WindowSlide>>`. Une fenêtre peut être de type temporel `<<LevelTemporalitySlide>>`, et de type "range" de tuples `<<LevelRangeSlide>>`. `<<LevelTemporalitySlide>>` possède deux tagged values représentant la fréquence de la requête `<<frequency>>` et la taille de la fenêtre `<<size>>`. En particulier, `<<frequency>>` est associé à une tagged value `<<granularity>>` qui représente la granularité temporelle (par exemple minute, heure, jour, etc.). Comme pour les indicateurs non continus les niveaux d'agrégation sont associés à des données externes modélisées en utilisant le package `<<Externaldata>>`. Un exemple est montré à la Figure 5. Ici la requête Q1 est implémentée par l'indicateur `SommeEnergieTemporelle`. La mesure appelée value est agrégée avec trois fonctions d'agrégation SUM, MIN et MAX. Ensuite seconds définit la fenêtre temporelle sur les 3 dernières secondes et la requête est exécutée toutes les 3 secondes. Notons aussi qu'un autre indicateur `SommeEnergieFenetreTuples` est défini. Il permet la somme, le min et le max des 5 dernières relevés d'énergie arrivés.

Profil UML BI2

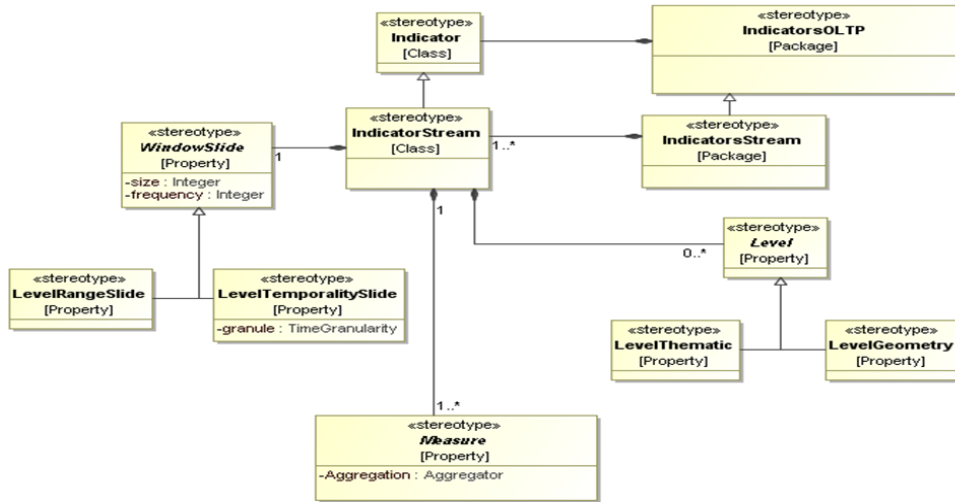


FIG. 4 – BI2 UML profil : IndicatorStream

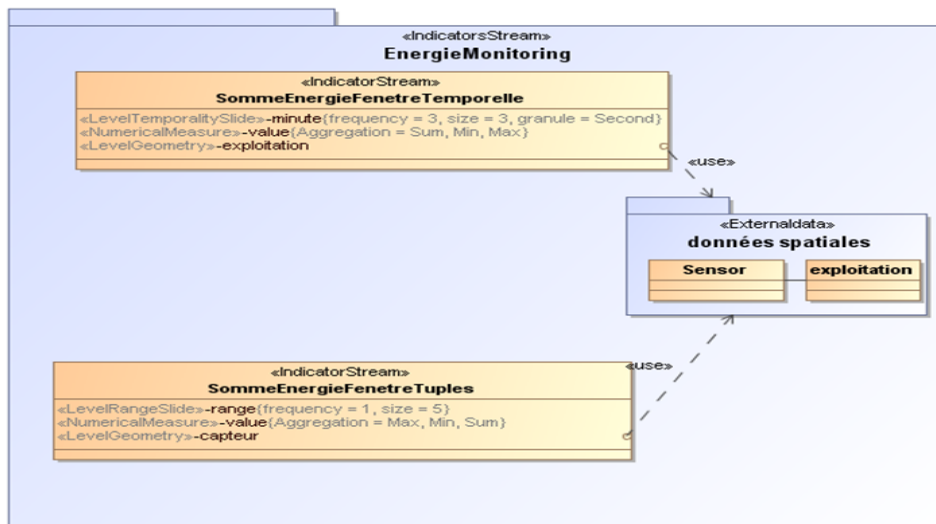


FIG. 5 – BI2 UML profil : IndicatorStream Q1

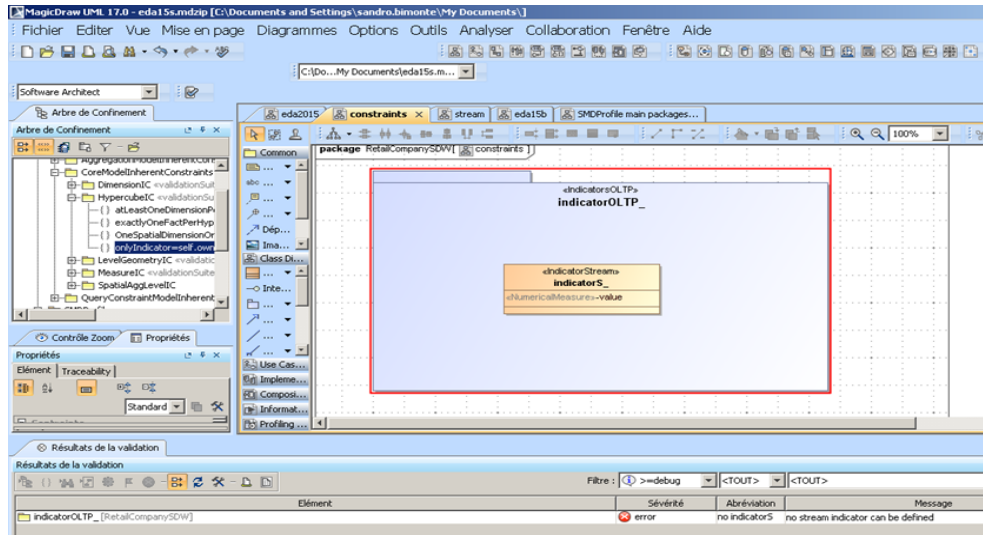


FIG. 6 – B12 UML profil : implementation en MagicDraw

4.4 AGL : MagicDraw

Le profil UML est implémenté dans l'AGL MagicDraw (MagicDraw2014). MagicDraw permet la définition de profils qui étendent les éléments du méta-modèle d'UML et implémente des contraintes OCL. Lors de la conception de modèles, ces contraintes peuvent être ensuite contrôlées afin d'éviter la création de modèle incorrect. Un exemple de contrainte OCL qui définit qu'un package « IndicatorsOLTP » ne doit pas contenir d'indicateur stream (« IndicatorStream ») est : `Context IndicatorsOLTP inv: self.ownedMember->select (m | m.ocIsTypeOf (IndicatorStream))->size () = 0` Comme montré en Figure 6, MagicDraw vérifie cette contrainte lors de la définition des indicateurs et signale une erreur si elle n'est pas satisfaite.

5 Implémentation

Dans cette section nous présentons l'implémentation des indicateurs OLTP (Sec 5.1) et Stream (Sec 5.2).

5.1 Indicateur OLTP : SQL

Dans cette section nous présentons l'implémentation des indicateurs OLTP. Ceux-ci sont implémentés en utilisant comme source, des données stockées dans un SGBD relationnel (comme Postgres par exemple), via le template de requête SQL présenté en Figure 7. Il s'agit d'une requête SQL de type 'Aggregation : group by'.

`Aggregation (Indicator.Measurei.value)` représente la *i*-ème mesure à agréger avec sa fonction d'agrégation de la classe « Indicator ». `Externaldataj` représente une table

Profil UML BI2

```
SELECT Aggregation(Indicator.Measure1.value),
...,Aggregation(Indicator.Measure_n.value),
Externaldata_2_Level_2, ...,Externaldata_m_Level_m

FROM Externaldata_measure_1, ...,Externaldata_m
WHERE (Externaldata_measure_1.att_measure = Externaldata_2.att AND
...
AND (Externaldata_measure_m.att_measure = Externaldata_m.att_m))

GROUP BY Externaldata_2_Level_2, ..., Externaldata_m_Level_m
```

FIG. 7 – *Implementation IndicatorOLTP en SQL*

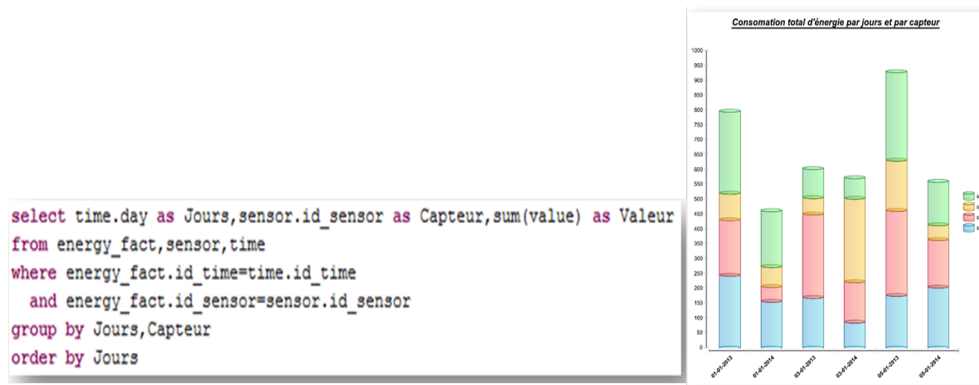


FIG. 8 – *Implémentation de l'indicateur OLTP de la figure 3 avec sa visualisation*

ou une vue contenant un attribut avec le nom du niveau d'agrégation `Level_j` ou de la mesure `Measure_i` (`Externaldatameasure`). Le reste de la requête doit présenter une jointure entre ces tables/vues et le `group by` sur les niveaux d'agrégation `Level_2 . . . Level_m`. L'implémentation de l'indicateur de la Figure 3 est montrée en Figure 8 avec sa visualisation dans l'outil de reporting SpagoBI.

5.2 Indicateur Stream : Esper

Pour l'implémentation des indicateurs stream nous avons utilisé Esper (Esper2014). Esper est un DSMS, se présentant sous la forme d'une API pour différents langages de programmation, comme par exemple Java. Le template de la requête qui implémente un indicateur stream est montré en Figure 9. Nous remarquons qu'une requête SQL pour chaque niveau de l'indicateur est définie, et que la fenêtre temporelle est définie dans le langage d'ESPER au sein de la définition de la clause FROM. L'indicateur `SommeEnergieTemporelle` est implémenté de la façon décrite à la Figure 10. Nous remarquons la définition de la fenêtre temporelle (`timebatch` (3 sec)) avec la somme, la requête SQL pour le `group by` par exploitation (Figure 10a), et le résultat qui s'affiche de façon continue (toutes les 3 secondes) (Figure 10b).

```

SELECT
Aggregation(Indicator.Measure1.value),...,Aggregation(Indicator.Measuren.value),
Externaldata2.Level2, ..., Externaldatam.Levelm

FROM Event.win:LevelTemporalitySlide| LevelRangeSlide as Event
+
"sql:Data2 ['select Externaldata2.Level2 from Externaldata2']as Externaldata2 " +
...
"sql:Datam ['select Externaldatam.Levelm from Externaldatam'] as Externaldatam "
WHERE (Eventmeasure.attmeasure = Externaldata2.att AND
...
AND (Eventmeasure.attmeasure = Externaldatam.attm )

GROUP BY Externaldata2.Level2, ..., Externaldatam.Levelm

```

FIG. 9 – Implementation IndicatorStream en Esper + SQL

```

"select labelfarm as labelfarm,sum(consomationStream) as sum_val from SensorEvent.win:time_batch(3 sec) as sensorevent," +
"sql:streamOlap['select labelfarm,farm.id_farm as farm,sensor.id_farm as s_farm,id_sensor from farm,sensor']as sensorbdfarm " +
"where sensorevent.label=sensorbdfarm.id_sensor " +
"and sensorbdfarm.s_farm=sensorbdfarm.farm " +
"group by labelfarm";

```

(a)

```

SensorEvent [Mon Dec 01 16:18:38 CET 2014 : s4 consumed 225]
SensorEvent [Mon Dec 01 16:18:38 CET 2014 : s5 consumed 139]
SensorEvent [Mon Dec 01 16:18:38 CET 2014 : s5 consumed 51]
SensorEvent [Mon Dec 01 16:18:38 CET 2014 : s2 consumed 136]
SensorEvent [Mon Dec 01 16:18:39 CET 2014 : s3 consumed 155]
SensorEvent [Mon Dec 01 16:18:39 CET 2014 : s7 consumed 256]
SensorEvent [Mon Dec 01 16:18:39 CET 2014 : s4 consumed 87]
SensorEvent [Mon Dec 01 16:18:39 CET 2014 : s6 consumed 170]
SensorEvent [Mon Dec 01 16:18:39 CET 2014 : s4 consumed 20]
SensorEvent [Mon Dec 01 16:18:40 CET 2014 : s5 consumed 93]
SensorEvent [Mon Dec 01 16:18:40 CET 2014 : s5 consumed 251]
SensorEvent [Mon Dec 01 16:18:40 CET 2014 : s7 consumed 255]
SensorEvent [Mon Dec 01 16:18:40 CET 2014 : s5 consumed 121]
-----
- [MONITOR] ValueSUM of | farm3 ----> 170
-----
- [MONITOR] ValueSUM of | farm2 ----> 866
-----
- [MONITOR] ValueSUM of | farm1 ----> 935
-----
SensorEvent [Mon Dec 01 16:18:41 CET 2014 : s2 consumed 230]
SensorEvent [Mon Dec 01 16:18:41 CET 2014 : s2 consumed 280]
SensorEvent [Mon Dec 01 16:18:41 CET 2014 : s3 consumed 51]
SensorEvent [Mon Dec 01 16:18:41 CET 2014 : s5 consumed 177]
SensorEvent [Mon Dec 01 16:18:42 CET 2014 : s7 consumed 38]
SensorEvent [Mon Dec 01 16:18:42 CET 2014 : s2 consumed 167]
SensorEvent [Mon Dec 01 16:18:42 CET 2014 : s3 consumed 294]
SensorEvent [Mon Dec 01 16:18:42 CET 2014 : s2 consumed 228]
SensorEvent [Mon Dec 01 16:18:43 CET 2014 : s2 consumed 191]
SensorEvent [Mon Dec 01 16:18:43 CET 2014 : s1 consumed 275]
SensorEvent [Mon Dec 01 16:18:43 CET 2014 : s6 consumed 78]
SensorEvent [Mon Dec 01 16:18:43 CET 2014 : s4 consumed 245]
SensorEvent [Mon Dec 01 16:18:43 CET 2014 : s6 consumed 18]
-----
- [MONITOR] ValueSUM of | farm3 ----> 108
-----
- [MONITOR] ValueSUM of | farm2 ----> 546
-----
- [MONITOR] ValueSUM of | farm1 ----> 1736

```

(b)

FIG. 10 – Implementation IndicatorStream : Q1

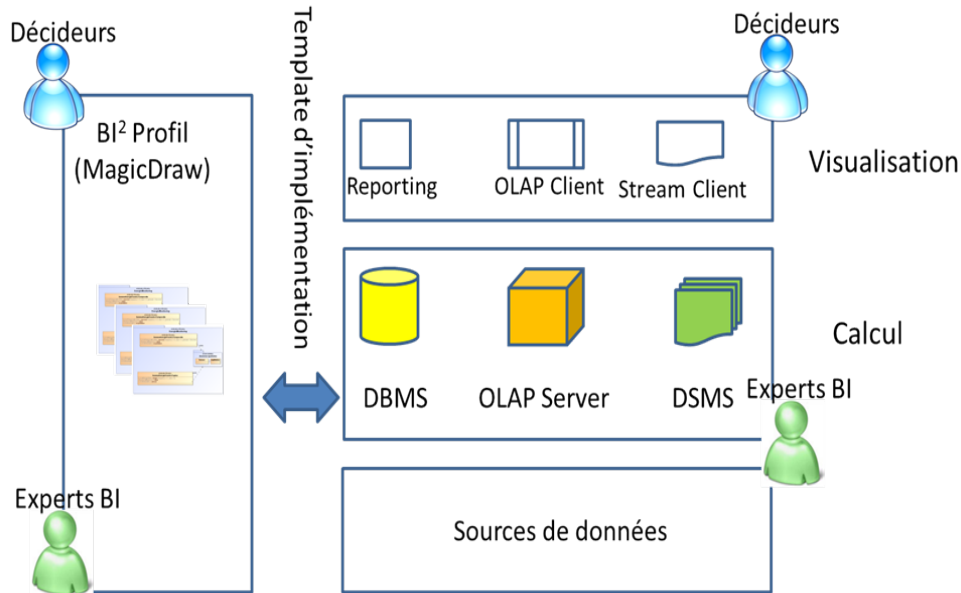


FIG. 11 – Vue d'ensemble

5.3 Vue d'ensemble

Dans cette section nous présentons une vue d'ensemble du système BI qui peut être mis en place pour notre cadre conceptuel (Figure 11). Les experts BI et les décideurs peuvent définir ensemble en utilisant le profil BI2 dans MagicDraw l'ensemble des indicateurs décisionnels qui doivent être implémentés. Une fois ceux-ci définis, les experts BI peuvent créer les requêtes qui implémentent les indicateurs OLTP et Stream en suivant les templates de requêtes décrits en sections 5.1 et 5.2, qui utiliseront donc les sources de données, et un DBMS pour les indicateurs OLTP et un DSMS pour les indicateurs Stream. Pour les indicateurs OLAP un modèle multidimensionnel implémenté dans un serveur OLAP est utilisé selon l'approche décrite en (Boulil et al., 2015). Ensuite, les résultats de ces indicateurs, sous forme d'affichages graphiques, sont analysés par les décideurs. Dans le cadre que nous proposons, il est important de mettre en évidence que cette complexité architecturale d'implémentation est transparente en phase de conception grâce à notre profil UML.

6 Travaux existants

Différents travaux se sont intéressés aux modèles conceptuels pour les EDs. En particulier, les aspects temporels ont été introduits au niveau conceptuel par (Malinowski et Zimányi, 2008) qui définissent un modèle conceptuel pour les entrepôts de données temporelles. Les auteurs introduisent des types temporels dans les niveaux, les hiérarchies et les mesures. En particulier, ils définissent quatre types temporels : temps valide (VT), temps de transaction (TT), le temps de chargement (LT), et durée de vie (LS) pour indiquer respectivement la pé-

riode pendant laquelle les données sont valables dans la réalité, la période pendant laquelle les données sont présentes dans les sources de données, la période pendant laquelle les données sont présentes dans l'entrepôt de données, et la période pendant laquelle des données existent dans la réalité. En ce qui concerne les systèmes d'information en temps réel, les auteurs de (Douglass, 1998) définissent un profil UML pour les types de données en temps réel. En conclusion la modélisation conceptuelle de données en flot n'a pas été abordée comme souligné par (Bimonte et al., 2012). En ce qui concerne la modélisation des indicateurs décisionnels, plusieurs modèles conceptuels ont été proposés pour leur définition d'un point de vue utilisateur/décideurs (Pardillo et al., 2011) (Giorgini et al., 2008), mais ces modèles reposent tous sur une modélisation multidimensionnelle. Plus globalement, les auteurs se sont intéressés à l'expression des besoins fonctionnels et aux structures des bases de données (stream) (i.e. OLTP, DSMS), ainsi qu'au processus qui lie les données OLTP et OLAP (ETL) (Akkaoui et al., 2012), en laissant la représentation de leur interrogation aux langages textuels comme SQL pour les systèmes OLTP, ou alors à des langages visuels qui définissent les opérateurs d'interrogation en se basant sur des métaphores visuelles (Bonhomme et al., 1999). Dans le contexte des entrepôts de données, des travaux définissent des langages d'interrogation tel que MDX ainsi que des langages visuels, mais seule la structure de l'ED définit l'ensemble des indicateurs décisionnels.

7 Conclusions

De plus en plus de données sont disponibles pour une analyse décisionnelle via des indicateurs décisionnels différents. Bien que différentes technologies décisionnelles aient été développées, nous constatons un manque de cadre conceptuel pour la définition et l'implémentation de ces indicateurs. Dans ce papier, nous présentons une première classification de ces indicateurs. De plus, motivés par le besoin d'un formalisme pour la définition de ces indicateurs à un niveau conceptuel, nous présentons un profil UML BI2 unique qui permet de représenter des indicateurs OLAP, OLTP et stream. Nous présentons également leur implémentation dans des outils industriels existants. Actuellement, nous travaillons sur les indicateurs de type *Données non Historiques*, *Fréquence continue*, *Navigation hiérarchique*, et sur la génération automatique des requêtes d'implémentation. L'identification des indicateurs dépend des besoins décisionnels collectés. L'identification de ces indicateurs peut être une tâche délicate pour le concepteur. L'évolution de ces besoins apporte de plus une difficulté supplémentaire. C'est pourquoi, nous travaillons sur la définition d'une méthodologie de conception orientée besoins d'analyse, comme dans les EDs (Giorgini et al., 2008), pour la conception de ces différents indicateurs.

Références

- Akkaoui, Z. E., J. Mazon, A. A. Vaisman, et E. Zimanyi (2012). Bpmn-based conceptual modeling of etl processes. In *Data Warehousing and Knowledge Discovery - 14th International Conference, DaWaK 2012, Vienna, Austria, September 3-6, 2012. Proceedings*, pp. 1–14.
- Andrienko, N. V., G. L. Andrienko, et P. Gatalsky (2003). Exploratory spatio-temporal visualization : an analytical review. *J. Vis. Lang. Comput.* 14(6), 503–541.

Profil UML BI2

- Bimonte, S., O. Boussaid, et L. E. Sarraj (2012). New conceptual modeling requirements for stream data warehouses (C). In *Actes du XXXeme Congrès INFORSID, Montpellier, France, 29 - 31 mai 2012*, pp. 463–472.
- Bimonte, S., J. Capdeville, et Y. Lefrileux (2014a). Energetic assessment of dairy activities using olap systems. *DEU Ageng 2014 International conference on agricultural engineering*, 10.
- Bimonte, S., J.-P. Chanet, J. Capdeville, A. Tailleur, et M. Luciano (2014b). Une etude sur l'efficacite des methodes de conception et implementation pour les entrepots de donnees par une methodologie "requirement-based" : Cas d etude de la consommation d energie en agriculture. *Revue des Nouvelles Technologies de l'Information Xe journees francophones sur les Entrepots de Donnees et Analyse en Ligne, RNTI-B-10*, 119–128.
- Bonhomme, C., C. Trepied, M.-A. Aufaure-Portier, et R. Laurini (1999). A visual language for querying spatio-temporal databases. In *ACM-GIS '99, Proceedings of the 7th International Symposium on Advances in Geographic Information Systems, November 2-6, 1999, Kansas City, USA*, pp. 34–39.
- Boulil, K., S. Bimonte, et F. Pinet (2015). Conceptual model for spatial data cubes : A uml profile and its automatic implementation. *Computer Standards and Interfaces* 38(0), 113 – 132.
- Chen, H., R. H. L. Chiang, et V. C. Storey (2012). Business intelligence and analytics : From big data to big impact. *MIS Quarterly* 36(4), 1165–1188.
- Cuzzocrea, A., L. Bellatreche, et I. Song (2013). Data warehousing and OLAP over big data : current challenges and future research directions. In *Proceedings of the sixteenth international workshop on Data warehousing and OLAP, DOLAP 2013, San Francisco, CA, USA, October 28, 2013*, pp. 67–70.
- Douglass, B. P. (1998). *Real-time UML - developing efficient objects for embedded systems*. Addison-Wesley object technology series. Addison-Wesley-Longman.
- Giorgini, P., S. Rizzi, et M. Garzetti (2008). Grand : A goal oriented approach to requirement analysis in data warehouses. pp. 4–21.
- Golab, L. et M. T. Ozsu (2010). *Data Stream Management*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers.
- Malinowski, E. et E. Zimányi (2008). A conceptual model for temporal data warehouses and its transformation to the ER and the object-relational models. *Data Knowl. Eng.* 64(1), 101–133.
- Mazon, J.-N. et J. Trujillo (2008). An mda approach for the development of data warehouses. *Decision Support Systems* 45(1), 41–58.
- Middelfart, M. et T. B. Pedersen (2010). Using sentinel technology in the TARGIT BI suite. *PVLDB* 3(2), 1629–1632.
- Pardillo, J., J.-N. Mazon, et J. Trujillo (2011). An mda approach and qvt transformations for the integrated development of goal oriented data warehouses and data marts. *J. Database Manag.* 22(1), 43–68.
- Torlone, R. (2003). Conceptual multidimensional models. In *Multidimensional Databases*, pp. 69–90.

Summary

Today more and more data are available for analysis and decision-making based on different indicators. Although different BI technologies have been developed, we find the lack of a conceptual framework for the definition and implementation of these indicators. In this paper, we present a first classification of these indicators. In addition, motivated by the need for a formalism for the definition of these indicators at a conceptual level, we present a UML profile that allows BI2 represent OLAP indicators, OLTP and streaming. We also present their implementation in existing industrial tools.

