Une méthodologie collaborative de conception d'entrepôt de données

Amir Sakka, Sandro Bimonte, Lucile Sautot, Guy Camilleri, Pascale Zaraté, Aurelien Besnard

Une Méthodologie Collaborative de Conception d'Entrepôt de Données

Amir Sakka*,**, Sandro Bimonte*, Lucile Sautot***, Guy Camilleri**, Pascale Zaraté**, Aurélien Besnard****

* IRSTEA, UR TSCF, 9 Av. B. Pascal, 63178, Aubière, France
amir.sakka@irstea.fr, sandro.bimonte@irstea.fr
** Université Paul Sabatier, IRIT, Toulouse, France
guy.camilleri@irit.fr, pascale.zarate@irit.fr
*** AgroParistech, UMR TETIS, 500 rue Breton, Montpellier, France
lucile.sautot@agroparistech.fr
**** LPO Aquitaine, 433 Chemin de Leysotte, 33140 Villenave-d'Ornon, France

Résumé. Dans le contexte de l'information géographique volontaire (VGI), les volontaires ne sont pas impliqués dans les processus décisionnels. De plus, les systèmes VGI n'offrent pas d'outils puissants pour mener des analyses temporelles. C'est pourquoi, dans cet article, nous proposons d'utiliser les systèmes d'information décisionnels pour analyser les données VGI, et nous proposons la définition une nouvelle méthodologie de conception des entrepôts de données, qui permet l'implication des volontaires dans la définitions des besoins analytiques sur les données VGI. Nos propositions ont été testées sur un cas d'étude réel concernant la biodiversité.

1 Introduction

Les dispositifs de science collaborative (ou science citoyenne) sont définis comme des modèles de production de données pour lesquels la résolution des problèmes est distribuée entre les utilisateurs via un système en ligne [Brabham (2008)]. Wikipedia [1], les forums, sont des exemples bien connus de systèmes collaboratifs. Dans les systèmes collaboratifs, les utilisateurs ajoutent, suppriment et modifient les contenus (ex : réponses de forums, documents, etc.) jusqu'à ce que la communauté arrive à un accord. Concernant les données géographiques, le "*crowdsourcing*" a été défini comme l'information géographique volontaire (VGI). Le VGI est la mobilisation d'outils pour créer, assembler, et disséminer les données géographiques fournies par des volontaires [Sui et al. (2012)]. Le VGI permet de gérer de données géolocalisées (un exemple : Openstreetmap [2]), et est largement utilisé dans différents domaines d'application comme l'urbanisme, la biodiversité ou encore la gestion des risques. Classiquement, les volontaires sont des producteurs de données actifs, et des consommateurs passifs des analyses de données fournies par les organismes ou les entreprises concernées. Ce paradigme, avec

1. https ://www.wikipedia.org
2. https ://www.openstreetmap.org

un processus de création de données "*bottom-up*" et avec un processus d'analyse de données "*top-down*", représente une barrière importante pour le développement d'observatoires basés sur des contributions volontaires, car les producteurs de données peuvent se sentir exclus du processus de prise de décision [Levrel et al. (2010)]. De plus, comme souligné dans [Bimonte et al. (2014)], le VGI ne propose pas de fonctionnalités analytiques pour de grands volumes de données spatiales. Dans [Bimonte et al. (2014)], l'analyse de données VGI grâce à la GeoBusiness Intelligence (GeoBI) a démontré son efficacité. Plus précisément, la VGI est conÃ§ue pour des tâches opérationnelles et des analyses spatiales complexes sur un petit jeu de données géographiques, alors que les systèmes OLAP Spatial (SOLAP), extension des systemes OLAP, sont pertinents pour les analyses basées sur l'exploration de grandes bases de données géographiques stockés dans un entrepôt de données spatial (SDW) [Stefanovic et al. (2000), Kimball et al. (2015)]. Comme les entrepôts de données sont conÃ§us en fonction des sources de données et des besoins des utilisateurs, plus le modèle de l'entrepôt de données reflète les besoins des parties prenantes, plus les parties prenantes utilisent les données [Kimball et al. (2015),Romero et Abelló (2009)]. Fournir des applications SOLAP correspondant aux besoins analytiques d'une communauté VGI représente une avancée sociale et économique importante, car : d'une part, des nouvelles possibilités d'analyse seront possibles, et d'autre part, les volontaires seront de plus en plus motivés pour la collecte de données. C'est pourquoi cet article a pour objectif de changer *les volontaires producteurs de données en analystes de données, grâce à un nouveau type de système OLAP*, décrit ci-dessous. Dans cet article nous nous interessons aux systemes OLAP en ignorant les aspects spatiaux des systemes SOLAP.

Notre proposition : Dans le même esprit que les méthodologies de validation de données adoptées par les systèmes collaboratifs existants (OpenStreetMap, Wikipedia, etc.), nous présentons dans la Figure 1 notre vision d'une nouvelle méthodologie de conception de système OLAP : OLAP 2.0. L'idée principale de notre proposition est de permettre aux volontaires d'exprimer leurs besoins analytiques lors d'une première étape. Ces besoins sont ensuite traduits en modèles multidimensionnels, c'est-à-dire en modèles conceptuels d'entrepôt de données. Ensuite, ces modèles sont proposés à un ensemble de volontaires particuliers appelés "*committers*", qui sont complètement impliqués dans le projet, et très expérimentés. [Kimball et al. (2015)] souligne la nécessité de mettre en place un "*data steward*" (conduite par les committers dans notre approche) afin de résoudre : (i) les problèmes relatifs au manque d'expérience des utilisateurs concernant les spécifications de requêtes (i.e. besoins d'analyse), et (ii) les problèmes relatifs à la propriété ou à la confidentialité des données qui apparaissent lors de l'implémentation d'un entrepôt de données. Par conséquent, les committers décident si les besoins recueillis auprès des volontaires (sous forme de modèles multidimensionnels) seront implémentés ou non, en jugeant de la pertinence de ces besoins, grâce à leur expertise dans le domaine d'application. Après cela, des informaticiens experts des systèmes d'information décisionnels implémentent les modèles approuvés par les committers (Figure 1). Pour finir, les nouveaux modèles multidimensionnels implémentés sont mis à disposition de tous les utilisateurs, qui peuvent ainsi visualiser, explorer et analyser les données (Figure 1).

Problématique de recherche : Dans cet article, nous nous concentrons sur la problématique posée par la conception collaborative de modèles multidimensionnels. Cependant, nous ne traiterons pas les problèmes relatifs à la qualité des données recueillies par des systèmes collaboratifs : nous appliquerons la méthodologie proposée sur des bases de données VGI dont

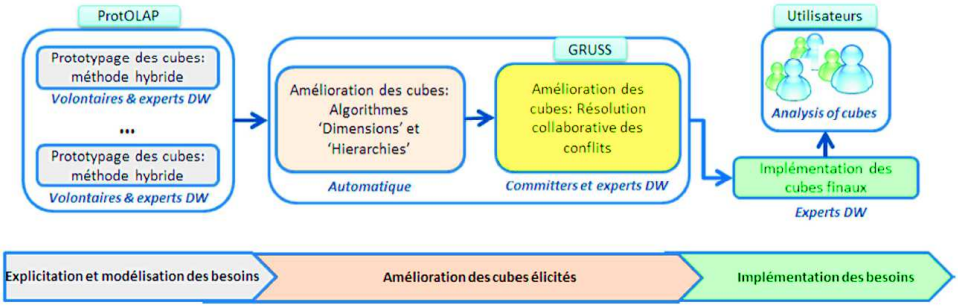

FIG. 1: La méthodologie OLAP 2.0.

la qualité est parfaitement maîtrisée, le processus de nettoyage des données ayant été défini et appliqué dès la collecte des données. Plusieurs méthodologies de conception d'entrepôts de données ont été proposées dans la littérature [Cravero et Sepúlveda (2014),Romero et Abelló (2009)]. Néanmoins, quand les décideurs sont des volontaires et ne font pas partie des "committers", ils ont les caractéristiques suivantes :

1. ils représentent seulement une petite partie des utilisateurs potentiels du système OLAP, ainsi, leur besoins analytiques ne sont peut-être pas ceux qui seront les plus utiles aux utilisateurs finaux ;
2. ils peuvent avoir des parcours et des fonctions différentes (scientifiques, citoyens, etc.), ce qui peut conduire à multiples interprétations contradictoires des mêmes besoins. Quand les parties prenantes ont des buts différents, il devient difficile de maintenir un accord entre elles, comme le montre les travaux en ingénierie des besoins [Egyed et Grunbacher (2004),Pohl (2010)] ;
3. ils n'ont pas de compétences particulières concernant les entrepôts de données, voire concernant les systèmes d'information : il est donc possible qu'ils ne formalisent pas correctement ou pas clairement leurs besoins ;
4. ils peuvent être nombreux, ce qui rend la gestion des conflits particulièrement compliquée ;
5. ils ne sont pas employés par le projet, mais bénévoles : leur implication en terme de temps dans le projet est donc limitée, et ils ne peuvent pas, faute de temps, définir leurs spécifications de manière exhaustive et précise.

Ainsi, les besoins définis par ces utilisateurs présentent [Pohl (2010)] : des **similarités** , c'est-à-dire, des éléments multidimensionnels identiques qui ont été définis de la même manière plusieurs fois indépendamment ; des **différences**, c'est-à-dire, différentes définitions pour des éléments multidimensionnels identiques ; des **conflits**, c'est-à-dire des définitions d'éléments multidimensionnels soit erronées, soit qui ne sont pas pertinentes.

De fait, impliquer ces utilisateurs particuliers dans une méthodologie de conception d'entrepôt de données existantes n'est pas possibles car ces méthodologies : a) nécessitent une connaissance avancée des principaux concepts de la modélisation multidimensionnelle (voir

3 ci-dessus) ; b) font l'hypothèse que les utilisateurs sont effectivement impliqués dans l'ensemble du projet, ce qui permet à leurs besoins d'analyse d'être justes et completement définis (voir 1, 2 et 5 ci-dessus) ; c) peuvent générer un nombre important de modèles (voir 2 et 4 ci-dessus) avec une implémentation coûteuse ; d) peuvent considérer des analyses erronées au vu du contexte applicatif (voir 2, 4 et 5 ci-dessus).

Pour résoudre ces questions, en se basant sur les principes de l'ingénierie des besoins [Pohl (2010)], nous proposons une méthodologie innovante de conception collaborative d'entrepôt de données, utilisant un système d'aide à la décision en groupe (GDSS), qui aide les committers dans la décision d'implémenter ou pas les besoins d'analyse recueillis auprès des volontaires. En effet, les outils de GDSS sont conçus pour aider un groupe engagé dans un processus de décision collectif et collaboratif. Ce type de système a été utilisé dans différents domaines tels que les workflows, les interfaces utilisateurs ou encore les bases de données [Zaraté (2013)], mais n'a pas encore été utilisé pour la conception d'entrepôt de données. De plus, pour permettre aux volontaires d'éliciter facilement leurs besoins (Figure 1), nous utilisons la méthodologie ProtOLAP [3], qui propose un prototypage rapide d'entrepôt de données, afin de faciliter sa conception avec les utilisateurs. Pour valider nos propositions, nous utiliserons le cas d'étude du projet ANR VGI4Bio. L'article est organisé comme suit : la Section 2 présente le cas d'étude sur lequel nous travaillons, la Section 3 présente la méthodologie proposée, la Section 4 montre l'implémentation et la validation des propositions et la Section 5 décrit l'état de l'art.

2 Cas d'étude

Pour les besoins du projet VGI4Bio [3], nous avons mobilisé deux bases de données VGI (Visionature et l'Observatoire Agricole de la Biodiversité [4] - OAB) afin de mettre en place une application SOLAP, qui permettra l'analyse d'indicateurs de biodiversité dans les milieux agricoles. Visionature et l'OAB impliquent respectivement 7682 et 1500 volontaires qui collectent les données. Parmi les utilisateurs qui seraient intéressés par l'analyse de ces données, nous avons identifié un grand nombre de personnes qui peuvent être réparties en différentes catégories : les volontaires eux-mêmes, qui souhaitent améliorer la qualité de leur production de données, mais également des organismes publics et privés (DREAL, Chambre d'Agriculture, etc.). A cette étape du projet, nous avons identifié quelques volontaires et un ensemble de committers. La Figure 2 montrent les modèles multidimensionnels définis par trois volontaires différents concernant l'analyse de l'abondance de plusieurs espèces animales. Ces modèles permettent de formuler des requêtes comme : "Quel est l'abondance totale des oiseaux selon l'altitude, l'espèce et la semaine ?" (Figure 2a). Ces modèles représentent les besoins analytiques des volontaires. D'une part, comme dans les méthodologies classiques de conception d'entrepôt de données, ces besoins peuvent présenter :

— des similarités comme "abondance+SUM", "jour", etc.

— des différences comme "Saison_bio", "comportement", etc.

D'autre part, puisque des volontaires différents, avec des objectifs différents, ont défini ces besoins, les différents modèles multidimensionnels proposés par les volontaires peuvent présenter des conflits. En effet, certains éléments multidimensionnels, considérés comme nécessaires par les utilisateurs lors de la définition du modèle, ne sont pas cohérents vis à vis

3. www.vgi4bio.fr
4. www.observatoire-agricole-biodiversite.fr

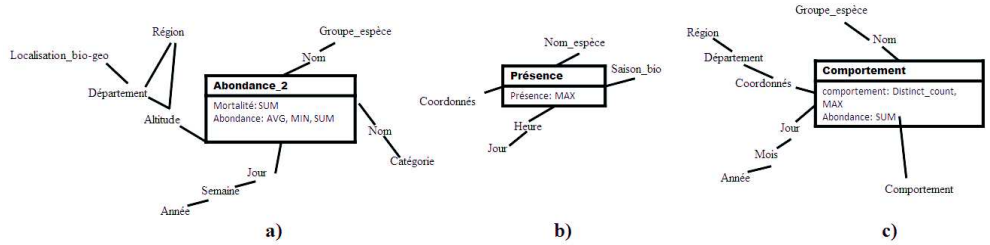

FIG. 2: Les modèles SOLAP des volontaires.

du domaine d'application. Par exemple, pour la mesure de l'abondance de certaines espèces ("abondance") comme les papillons, le protocole d'acquisition prévoit que l'opérateur observe les animaux en se déplaçant le long d'une ligne, d'une distance donnée, ou pendant une durée définie (environ 10 mètres pour les papillons). Ainsi, la mesure de l'abondance des papillons n'a de sens pour l'analyse biologique, que si elle est accompagnée de la durée ou de la longueur de l'observation. Ces conflits ne sont pas causés par les données sources, mais sont le résultat de disparités de connaissance et d'expertise du domaine d'application. Ainsi, ces conflits ne peuvent pas être résolus par un outil automatique, mais seulement par des committers compétents. De plus, en raison du grand nombre de volontaires impliqués dans un projet VGI, il est quasiment impossible de proposer une implémentation pour chaque modèle multidimensionnel proposé, en raison des coûts humain et économique de l'implémentation des systèmes OLAP. C'est pourquoi nous proposons de concevoir un ou plusieurs modèles multidimensionnels qui représentent un accord entre tous les volontaires, en résolvant les similarités, les différences et les conflits.

3 OLAP 2.0

Dans cette Section, nous définissons les étapes principales de notre méthode de conception collaborative d'entrepôt de données (Figure 1). Dans le reste de cet article, nous utiliserons les termes "besoin" et "modèle" respectivement pour "besoins analytiques" et "modèle multidimensionnel". La méthodologie proposée est composée des étapes suivantes :

1. Eliciation, modélisation et validation des besoins. Cette étape vise à collecter les besoins de chaque volontaire, de les traduire en modèles, puis de les valider en les confrontant aux données sources (voir paragraphe 3.1). Les deux étapes suivantes ont pour objectif de résoudre les problèmes liés aux besoins décrits précédemment.

Puisque les besoins sont traduits en modèles validés lors de l'étape 1, ces deux étapes suivantes fournissent une amélioration des modèles issues de l'étape 1 :

2. Résolution des différences et des similarités au sein des besoins. Cette étape fusionne les différents modèles proposés par les volontaires afin de résoudre les problèmes de similarités et de différences entre modèles, et génère des modèles améliorés (voir paragraphe 3.2). Ces conflits sont des problèmes structurels des modèles, et donc qui peuvent être resolus automatiquement via des algorithmes. Cette phase se base sur les travaux existants sur l'intégration

et la conception des EDs. Dans cet article nous détaillons notre approche pour des raisons de compréhensibilité de la méthodologie dans sa globalité.

3. Résolution collaborative des conflits. Cette étape a pour objectif de résoudre les conflits identifiés entre les différents besoins par les committers, qui décident quels besoins sont implémentés. Ces conflits ne sont pas structurels, mais liés aux besoins d'analyse, et donc ils ne peuvent pas être résolus automatiquement. Ainsi, une nouvelle amélioration est appliquée (voir paragraphe 3.3).

4. Les modèles qui ont recueilli l'agrément des committers sont implémentés. Il faut souligner que l'étape de conception collaborative n'a pas été ajoutée au début du processus de conception pour deux raisons : (a) le manque d'outils de conception collaborative pour les entrepôts de données, (b) l'impossibilité d'atteindre un accord entre committers a priori, sans s'appuyer sur un modèle comme base de discussion.

Dans la suite de cet article, nous définissons : (i) un indicateur comme une mesure, c'est à dire un fait + une fonction d'agrégation ; (ii) un cube comme un modèle (dimensions et faits) ; (iii) une hiérarchie comme un graphe dirigé acyclique de niveaux avec une racine connectée à chaque feuille par un unique chemin (par exemple la dimension "localisation" dans les Figure 2a a trois différentes hiérarchies : $\{Altitude-->Region$, $Altitude-->Departement-->Region$ et $Altitude-->Departement-->Localisationbio-geo\}$).

3.1 Elicitation, modélisation et validation des besoins

Cette étape est composée de deux phases : la première est l'*élicitation des besoins*, et la deuxième est la *traduction des besoins en modèles multidimensionnels valides*.

Nous utilisons la méthodologie ProtOLAP et son outil associé [3] pour l'élicitation des besoins des volontaires. En accord avec les techniques d'élicitation des besoins décrites dans [Bimonte et al. (2014)], ProtOLAP met en place : des entretiens, des réunions de travail, et du prototypage. En particulier, grâce à ProtOLAP, les volontaires définissent leurs besoins analytiques pendant les réunions, en langage naturel, et grâce à des documents Word ou Excel [Nuseibeh et Easterbrook (2000)]. Après cela, les informaticiens spécialisés en systèmes OLAP transforment ces besoins en modèles multidimensionnels, grâce au profil UML ICSOLAP, qui a été implémenté dans l'outil commercial MagicDraw. Enfin, l'outil ProtOLAP génère un prototype de cube (modèle et données) à partir des besoins exprimés. Ce prototype de cube est utilisé dans un processus itératif pour aider les volontaires dans l'élicitation des besoins. Ainsi, cette étape génère des nouveaux modèles multidimensionnels (voir Figure 2).

Après cette phase d'élicitation des besoins, les cubes sont validés sur les sources de données par les informaticiens grâce à une méthodologie existante [Cravero et Sepúlveda (2014),Romero et Abelló (2009)]. Les informaticiens associent à chaque modèle une description fournie par ses concepteurs (cette spécification sera utilisée ensuite dans l'étape de résolution collaborative des conflits). Par exemple, pour le modèle présenté sur la Figure 2b, le volontaire à l'origine du modèle fournit la description suivante : "Ce modèle est utilisé pour l'analyse de la couverture spatio-temporelle des données VGI".

Il est importante de noter que en utilisant ProtOLAP, les besoins analytiques sont simplement représentés avec des tableaux croisés dynamiques issues des cubes prototypés, qui peuvent être automatiquement traduits en modèles (voir [Nabli et al. (2005)]). Ceci permet d'éviter l'utilisation de formalismes complexes pour la spécification/représentation des besoins et des modèles multidimensionnels. De plus, les volontaires connaissent très bien le jeu

de données puisqu'ils l'utilisent et/ou l'alimentent. Ainsi, ils peuvent facilement définir des indicateurs à partir des données sources, ce qui facilite la validation des besoins sur les données sources. Enfin, pour éviter les problèmes d'alignement de vocabulaire pendant la phase d'élicitation avec ProtOLAP, les informaticiens contraignent les volontaires à utiliser le même vocabulaire, en utilisant, quand cela est possible, le répertoire de MagicDraw [Bakillah et al. (2006)]. Par exemple, pour la dimension temporelle, les informaticiens obligent chaque volontaire à utiliser le même nom de dimension : "Temps". Pour conclure, *cette étape a comme entrées les besoins exprimés par chaque volontaire, et comme sorties un ensemble de modèles multidimensionnels qui sont validés sur les données sources*.

3.2 Résolution des différences et des similarités au sein des besoins

Cette étape a pour objectif de résoudre les différences et les similarités des besoins définis dans l'étape précédente. C'est pourquoi, cette étape permet l'amélioration des modèles en les fusionnant en utilisant l'algorithme "Dimensions", présenté dans la Figure 3a. Pour chaque mesure commune aux différents modèles obtenus après la première étape, l'algorithme intègre toutes les dimensions de modèles dans un seul modèle. Par ce moyen, quand un volontaire exprime le même sujet d'analyse (c'est-à-dire la mesure) qu'un autre volontaire, mais en utilisant les dimensions différentes, l'algorithme Dimensions fournit le même sujet d'analyse, enrichi avec les dimensions définies par les deux volontaires. Par exemple, le modèle "F1" de la Figure 4 est la fusion de deux cubes "abondance_2" et "comportement" en raison de leur mesure commune "Abondance" avec leur dimension commune "espèces" (une dimension est dite 'commune' si elle est définie de la même manière dans des modèles différents), leurs dimensions non-communes "Comportement" et "Utilisateur" (une dimension est dite 'non-commune' si elle n'est définie dans un seul modèle) et leur dimensions non-conformes "temps" et "Localisation" (une dimension est dite 'non-conforme' si elle est définie différemment dans différents modèles).

Dans le même esprit, l'algorithme "Hierarchies" (Figure 3b) a pour objectif de fournir aux volontaires toutes les hiérarchies possibles définies pour les dimensions.

L'algorithme "Hierarchies" fusionne toutes les hiérarchies en un seul graphe, et trouve ainsi tous les chemins possibles des noeuds feuilles vers les noeuds racines. Quand le graphe a de multiples feuilles, les informaticiens doivent en choisir un seul. En effet, même si dans l'implémentation il est possible avoir plusieurs niveaux plus fins, au niveau conceptuel un seul niveau est possible en accord avec les travaux existants. Par exemple, dans le modèle "F1" dans la Figure 4b, le niveau "Coordonnés" est considéré comme le niveau le plus bas (plus bas que le niveau "Altitude") de la hiérarchie enrichie de la dimension "Localisation".

Pour conclure, cette étape permet de proposer aux volontaires des dimensions et des hiérarchies utiles pour l'analyse de leurs mesures, qui ont été involontairement oubliées ou volontairement ignorés (c'est-à-dire les différences entre besoins), et d'utiliser, quand c'est possible, les mêmes éléments multidimensionnels (c'est-àdire les similarités entre besoins).

3.3 Résolution collaborative des conflits

L'objectif de cette étape est de résoudre les conflits engendrés par l'étape précédente, en améliorant encore *tous* les modèles obtenus précédemment (voir paragraphe 3.2) : *les éléments multidimensionnels (mesures, dimensions, hiérarchies) ajoutés lors des étapes précédentes*

Entrée tous les cubes C
Sortie CubesFinals une suite de cubes
*1:**M** = mesures **de** C;*
2:**pour** *m* **de** M **faire**
3: **Générer** nouveau cube CubeFusion;
4: **Ajouter** m à CubeFusion;
5: **soit** DimsCommunes = dimensions communes **de** cubes **avec** m;
6: **soit** DimsNonCommunes = Dimensions non Communes de cubes **avec** m;
7: **Ajouter** DimsCommunes **et** DimsNonCommunes **à** CubeFusion;
8: **soit** DimsNonConformes = Dimensions Non Conformes **de** cubes **avec** m;
9: **Pour** d **de** DimsNonConformes **faire**
10: **soit** H = hiérarchies **de** d;
11: **d = FusionHiérarchies(H)**;
12: **Ajouter** d à CubeFusion;
13: **Fin Pour**
14: **Ajouter** CubeFusion **to CubesFinals;**
15: **Fin Pour**
16: **Pour** *cubes Cs* **avec** dimensions Communes **Faire**
17:**Générer CubesFinal avec** dimensions Communes **et** toutes Mesures **de** Cs
18: **Ajouter CubesFinal à CubesFinals**
19:**Fin pour**
20:**retourner** Cf

(a) L'algorithme "Dimensions".

Entrée: hiérarchies h1, ..., hn
Sortie: Dimension d
1: G = **Union** (h1, ..., hn);
2 V = **tous** Bottoms **de** G ;
4: **si** taille(V) > 1 **alors**
5: *choisir vBottom parmi V*
6: **pour** n dans V_bottoms **faire**
7: créerLien (G, vBottom, n);
8: **fin pour**
9: **fin si**
10: d = Ø;
11: **pour** chemin P **dans** G **faire**
12: **ajouter** P **à** d
13: **fin pour**
14: **retourner d**

(b) L'algorithme "Hierarchies".

FIG. 3: Les algorithmes de fusion utilisés.

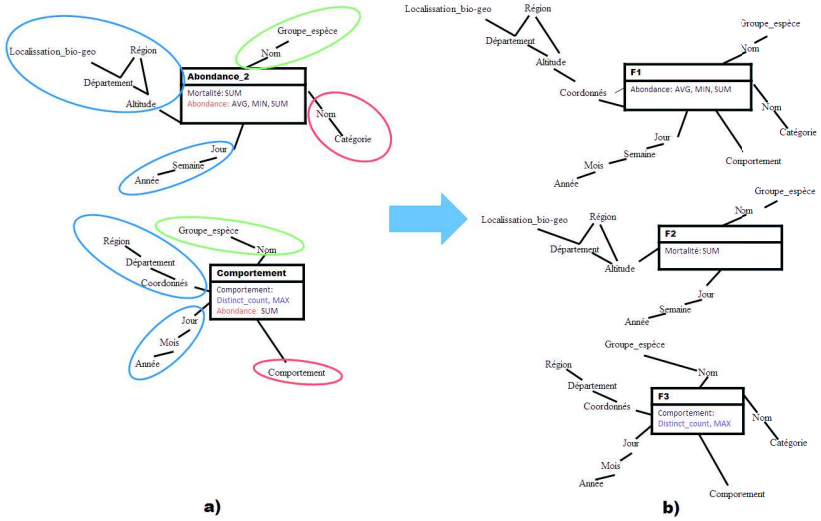

FIG. 4: Exemple de l'étape "Résolution des Différences est des Similarités".

sont-ils nécessaires ? L'amélioration est fournie par l'algorithme "*Collaborative design*" (Figure 5), grâce auquel les committers expriment leurs recommandations pour chaque élément multidimensionnel selon différents critères concernant leur utilité et leur utilisabilité. L'algorithme permet quand possible de trouver un consensus parmi les committers. Dans la suite de cette Section, nous décrirons l'algorithme, puis nous détaillons chaque méthode.

Algorithme de résolution collaborative des conflits au sein des besoins : en utilisant la méthode "NiveauExpertise", présenté dans le Tableau 1 (Figure 5, ligne 3), les committers définissent un niveau d'expertise pour le cube selon leurs compétences dans le domaine d'application. Ce niveau d'expertise permet de définir des prioritaires pour les choix des committers. Par exemple, un committer spécialiste de l'écologie peut définir son niveau d'expertise pour le cube "Comportement" (Figure 6a) comme "Elevé".

Après cela, les committers évaluent la pertinence analytique de chaque indicateur, dans le but de supprimer les indicateurs inutiles du cube final, grâce à la méthode "VoteIndicateurs", présenté dans le tableau 1 (Figure 5, ligne 4). Par exemple, tous les committers ont estimé que l'indicateur "Comportement+Min" n'est pas pertinent. Il est donc supprimé (Figure 6a).

Ensuite, si au moins un indicateur est conservé après la procédure de vote précédente, les committers évaluent la pertinence analytique de chaque dimension, dans le but de supprimer les dimensions inutiles, en utilisant la méthode "TriDimensions", présentée dans le tableau 1 (Figure 5, ligne 10). Par exemple, parmi les committers, le committer1 évalue comme pertinentes toutes les dimensions sauf la dimension "Utilisateur". Par conséquent, la dimension "Utilisateur" est supprimée (Figure 6b).

Seules les hiérarchies des dimensions conservées pendant cette étape seront évaluées à l'étape suivante. Notons que, dans le cas d'indicateurs holistiques (c'est-à-dire des indicateurs qui utilisent une fonction d'agrégation holistique, comme le distinct count [Kimball et al. (2015)]) sont supprimés quand une de leur dimension n'est pas conservée après la procédure de vote (Figure 5, ligne 11). En effet, les indicateurs de ce type risquent de ne plus pouvoir être calculés si l'accès au niveau le plus fin de granularité est interdit par la suppression de la dimension. Pour les autres indicateurs (les indicateurs distributifs et algébriques), l'élimination d'une dimension ne pose pas de problème puisque les mesures peuvent être agrégées sur le membre "ALL", puis réutilisées pour d'autres agrégations (comme des vues matérialisées [Kimball et al. (2015)]). Ainsi, dans notre exemple (Figure 6), puisque la dimension "Utilisateur" a été supprimée, l'indicateur holistique "comportement+distinct count" est également supprimé (Figure 7b). Une fois que toutes les dimensions jugées inutiles ont été supprimées, les committers évaluent les hiérarchies de chaque dimension, en se basant sur la richesse et la pertinence du plus bas niveau des hiérarchies (dans notre approche toutes les hiérarchies ont le même niveau plus bas par définition). Ceci est réalisé grâce à la méthode "VoteHiérarchies", présentée dans le tableau 1 (Figure 5, ligne 12). Pour notre cas d'étude, les committers ont décidé que toutes les hiérarchies de chaque dimension étaient bien définies. Notons que cette méthode élimine automatiquement une dimension si toutes ses hiérarchies sont considérées comme n'étant pas pertinentes. En effet, avec la méthode "VoteCubeDimensionutilisabilité", présentée dans le tableau 1 (Figure 5, ligne 15), les committers doivent évaluer l'utilisabilité du cube avec chaque dimension retenue [Golfarelli et Rizzi (2011)], car le nombre de dimensions utilisées affecte l'utilisabilité du cube, et donc le processus de prise de décision. Dans cette optique, l'algorithme, en commenÃ§ant par les dimensions les mieux notées par les com-

TAB. 1: Les méthodes de résolution collaborative des conflits des besoins.

Chaque committer	*Entrée*	*Sortie*	*Méthode*	*Critère*
NiveauExpertise	Cube	Niveau d'expertise	Auto-évaluation	Compétent dans le domaine d'application
VoteIndicateurs	Mesures	Mesures	Vote (Borda)	La mesure est utile
TriDimensions	Dimensions	Dimensions triées	Vote (Borda)	La dimension est utile
VoteHiérarchies	Hiérarchies et Dimensions	Dimensions triées	Multicritère (Sommes pondérées)	Les hiérarchies de la dimension sont assez riches La relation fait-dimension est pertinente
VoteCubeDimensionutilisabilité	Dimension	Dimension	Vote (Majoritaire)	Le cube avec la dimension est utilisable
VoteImplémentationCube	Cube	Cube final	Vote (Majoritaire)	Le cube doit être implémenté

mitters, ajoute des dimensions au cube consécutivement. Après chaque ajout de dimension, le cube résultant est présenté aux committers. Par ce moyen, les committers peuvent explorer le cube avec la dimension ajoutée, décider de l'utilisabilité du cube avec la nouvelle dimension, et donc choisir de la conserver ou pas. Enfin, les committers votent l'implémentation du cube résultant avec la méthode "VoteImplémentationCube", présentée dans le tableau 1 (Figure 5, ligne 18).

Description des méthodes : le tableau 1 présente les méthodes utilisées par l'algorithme de *résolution collaborative des conflits au sein des besoins d'analyse*.

"VoteIndicateurs" et "TriDimensions" utilisent une procédure de vote avec la méthode de calcul Borda [Gavish et Gerdes (1997)], puisqu'il n'y a qu'un critère de vote. La méthode "VoteHiérarchies" utilise une somme pondérée, car il s'agit d'une approche multicritère. Enfin, les méthodes "VoteCubeDimensionutilisabilité" et "VoteImplémentationCube" utilisent un vote majoritaire car un résultat booléen est attendu. A ce point, *le cube obtenu est composé uniquement de dimensions utilisables, utiles et bien formées, avec des indicateurs utiles*.

Dans ce qui suit, nous décrivons les différents critères utilisés par les méthodes. "La mesure est utile" et "La dimension est utile" [Chen et al. (2000)] sont utilisées pour évaluer la nécessité des indicateurs et des dimensions pour les objectifs analytiques des utilisateurs (i.e. sans ces indicateurs/ dimensions, les analyses produites via le cube sont fausses ou erronées).

Pour la méthode "VoteHiérarchies", puisqu'une analyse OLAP ne dépend pas seulement de la présence d'une dimension, mais surtout des niveaux des hiérarchies de cette dimension, nous avons défini les critères : (i) "La relation fait-dimension est pertinente" qui évalue si les faits sont présentés au bon niveau de granularité (c'est-àdire le niveau le plus bas de la dimension) et (ii) "Les hiérarchies de la dimension sont assez riches" qui évalue les possibilités analytiques offertes par la granularité de la hiérarchie. Le critère "Le cube avec la dimension est utilisable" [Chen et al. (2000), Golfarelli et Rizzi (2011)] est utilisé pour vérifier le degré d'utilisabilité du cube avec chaque dimension. Enfin, le critère "Le cube doit être implémenté" correspond à l'évaluation de la satisfaction des utilisateurs vis à vis du cube obtenu [Chen et al. (2000)]. Nous avons utilisé une échelle de notation à cinq niveaux.

Entrée: Cube c
Sortie Cube c
1: **Soit** I suite des indicateurs de c;
2: **Soit** D suite de dimensions de c;
3: NiveauExpertise(c)
4: I'=VoteIndicateurs(I);
5: **si** I' est vide **alors sortir;**
6: **pour** indicateur i de I non dans I' **faire**
7: **supprimer** i de c;
8: **fin pour**
9: **fin si**
10: D'=TriDimensions(D); //trier les dimensions
11 NettoyerIndicateurs (D', D) ;
12: D''=VoteHiérarchies(D');
13: NettoyerIndicateurs (D'', D') ;
14: **pour** dimension d de D'' **faire**
15: flag=VoteCubeDimensionUtilisabilité (d);
16: **si** flag est faux **alors** NettoyerIndicateurs (d, D'')
17: **fin pour**
18: VoteImplémentationCube(c);
191: **retourner** c;

Procédure NettoyerIndicateurs
Entrée suite de Dimension D' **sous-ensemble** de D
1: **pour** dimension d de D **non dans** D' **faire**
2: **supprimer** d de c;
3: **supprimer** tous indicateurs holistiques de c;
4: **fin pour**

FIG. 5: L'algorithme de résolution collaborative des conflits des besoins.

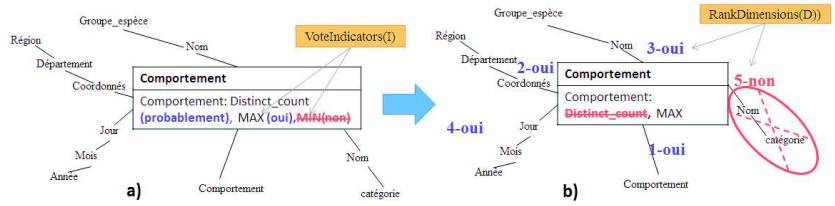

FIG. 6: Exemple de résolution collaborative des conflits des besoins.

4 Implémentation et validation

4.1 Implémentation

La méthodologie a été implémentée avec une architecture Relational OLAP classique, composée de PostgreSQL [5] comme système de gestion de base de données, Mondrian [6] comme serveur OLAP et JRubik [7] comme client OLAP. Nous utilisons le système ProtOLAP pour la première étape de la méthodologie. ProtOLAP prend comme entrée un modèle UML défini avec le profil UML ICSOLAP [Bimonte et al. (2013)], qui est implémenté dans MagicDraw.

5. https ://www.postgresql.org/
6. https ://github.com/pentaho/mondrian
7. http ://rubik.sourceforge.net/jrubik/intro.html

ProtOLAP crée automatiquement les scripts SQL pour PostgreSQL (création des tables et insertion des données), ainsi qu'un fichier XML de configuration de Mondrian.

La conception collaborative est implémentée grâce au système GRUS [Zaraté (2013)]. Nous avons défini un processus spécifique de décision de groupe constitué de 5 étapes. Il faut souligner que GRUS est un système web qui peut permettre un processus de décision asynchrone. Ainsi, GRUS est bien adapté à nos committers, qui ne travaillent pas tous au même endroit, et pas tous en même temps.

4.2 Expérimentation et validation

Pour réaliser cette expérience, nous avons mobilisé cinq volontaires avec différentes compétences, et nous avons identifié quatre committers.

Pour la validation de la première étape (Section 4.1), utilisant le système ProtOLAP, nous avons compté le nombre de réunion entre les volontaires et les informaticiens, ainsi que leur durée. Le temps d'implémentation d'un modèle d'entrepôt de données avec ProtOLAP est négligeable, puisque cette tâche ne prend que quelques minutes. En moyenne, il y a eu 3 réunions par volontaire et chaque réunion durait une heure. Ainsi, nous pouvons conclure que quand le nombre de volontaires est petit, l'utilisation de la méthodologie ProtOLAP est possible. Néanmoins, quand le nombre de volontaires devient grand, une nouvelle méthodologie doit être proposée pour permettre aux volontaires de définir leurs modèles multidimensionnels eux-mêmes, sans l'intervention des informaticiens.

Pour valider notre proposition de méthodologie de résolution collaborative de conflits au sein des besoins (Section 3.3), nous considérons un cube défini par un utilisateur expert en ornithologie, qui correspond exactement à ses besoins. Nous avons modifié ce cube en y ajoutant des dimensions et des indicateurs que l'expert considère comme inutiles. Par ce moyen, nous obtenons un cube dégradé. Nous avons notamment ajouté une dimension "Utilisateur" et une mesure "Max comportement". Enfin, nous avons soumis ce cube aux committers, et nous avons testé si, en utilisant notre méthodologie, les committers retrouvent le cube original. Les expériences ont permis de valider la méthodologie, puisque : (1) la méthode "VoteIndicateurs" a effectivement classé "Max comportement" comme la mesure la moins importante ; (2) la méthode "TriDimensions" a éliminé la dimension "Utilisateur", qui a obtenu seulement 7,7% des votes. Le cube original est donc retrouvé à la fin de l'étape collaborative.

Dans GRUS, la méthode Borda n'élimine pas les alternatives, et nous avons choisi pour chaque méthode de vote un seuil d'élimination des éléments multidimensionnels. Par exemple, pour la méthode "TriDimensions", un seuil de 40% est utilisé, c'est à dire qu'une dimension obtenant moins de 40% des votes sera éliminée.

Enfin, puisque les committers ne sont pas employés à cent pour cent par le projet, et qu'ils ne peuvent pas dépenser trop de temps pour cette tâche, nous avons aussi évalué la durée totale du processus collaboratif. Ce processus a duré moins d'une heure, et a été résolu en une seule réunion.

5 Etat de l'art

La conception d'entrepôts de données a été abordée dans de nombreux travaux [Cravero et Sepúlveda (2014),Romero et Abelló (2009)]. Trois types d'approches ont été définies : (i)

les méthodes basées sur les spécifications des utilisateurs (approche user-driven), qui définissent le schéma de l'entrepôt de données en utilisant uniquement les besoins analytiques des utilisateurs ; (ii) les méthodes basées sur les données sources (approche data-driven), où le schéma multidimensionnel est automatiquement dérivé des sources de données ; (iii) les méthodes mixtes (approche hybride), qui est une fusion des approches data-driven et user-driven. Les méthodes mixtes sont largement reconnues comme les plus efficaces. Elles fournissent des mécanismes pour valider les besoins des utilisateurs sur les sources de données et créer un modèle multidimensionnel [Cravero et Sepúlveda (2014)]. Cependant, comme nous l'avons décrit dans la Section 1, elles ne sont pas appropriées dans le contexte VGI, car elles ne fournissent pas de support collaboratif pour la résolution des conflits des besoins. En effet, bien que la gestion des conflits pendant la phase d'élicitation des besoins ait été explorée dans plusieurs domaines [Egyed et Grunbacher (2004)], ces techniques n'ont pas été encore appliquées à la conception d'entrepôt de données. A notre connaissance, uniquement [Corr et Stagnitto (2011)] propose une méthodologie agile, basée sur des questionnaires, afin d'aider les utilisateurs à travailler ensemble à la conception d'un entrepôt de données. Mais cette approche n'est pas formalisée et n'est pas accompagnée d'un outil informatique. Les méthodologies de conception collaborative ont été adoptées par de nombreux domaines (comme par exemple pour les systèmes d'information géographique [Driedger et al. (2007)], pour aider à maintenir la cohérence sémantique, pour l'e-learning [Ehn (2008)], etc.), mais pas pour les systèmes d'information. Les outils informatiques existants pour collecter les besoins analytiques utilisent des formalismes complexes [Cravero et Sepúlveda (2014),Romero et Abelló (2009)], ou des langages de requêtes (SQL, MDX, etc.). Cependant, dans notre approche, qui se base sur ProtOLAP, nous utilisons la même approche que [Nabli et al. (2005)], qui formalise les besoins d'analyse avec des tableaux croisés dynamiques. Ceci nous permet d'utiliser les tableaux croisés dynamiques (c'est-à-dire des prototypes de cubes) pour représenter les besoins des utilisateurs, mais également pour les éliciter. En effet, concernant l'élicitation des besoins d'analyse dans le contexte des entrepôts de données, mis à part les approches "manuelles" (réunions, rapports, etc.), certains travaux existants proposent des outils automatiques pour traduire les besoins définis en langage naturel en modèles multidimensionnelles [Naeem et al. (2012)]. Néanmoins, ces approches exigent que les utilisateurs aient de très bonnes compétences en OLAP, ce qui n'est pas le cas de nos volontaires.

6 Conclusion et travaux futurs

Dans cet article, nous proposons une nouvelle méthodologie de conception collaborative d'entrepôt de données, qui permet d'impliquer des volontaires dans la définition des besoins analytiques concernant des données VGI. Cette méthodologie permet à des volontaires sans aucune compétence en OLAP de participer au processus de conception de ce type d'outil. La méthodologie proposée a été implémentée et validée sur un cas d'étude réel concernant la biodiversité dans les milieux agricoles. Nos travaux futurs porteront sur l'application de la méthodologie collaborative à la définition de nouvelles hiérarchies, ainsi qu'au test d'autres méthodes de prise de décision en groupe. De plus, avec ProtOLAP, les informaticiens spécialistes des systèmes d'information décisionnels doivent obligatoirement assister les volontaires lors du processus d'élicitation des besoins, ce qui n'est pas réaliste dans un scénario de crowdsourcing des besoins à large échelle. Ainsi, à l'avenir, nous travaillerons sur un langage visuel

basé sur une table pivot pour soutenir la phase d'élicitation des besoins. Nous proposerons une extension des critères utilisés par notre approche collaborative en se basant sur des métriques qualitatives définies pour la satisfaction des utilisateurs d'entrepôts de données (comme dans [Chen et al. (2000)]), ainsi que sur les métriques quantitatives (comme celles proposées dans [Golfarelli et Rizzi (2011)]). Enfin, la prise en compte des aspects spatiaux pour les systÃ¨mes SOLAP, et une étude sur la gestion de la communauté (qui sont les volontaires, les committers, etc.) seront investigués.

Remerciements

Ces travaux ont recus le soutien financier du projet ANR-17-CE04-0012. Nous remercions respectueusement le Pr. Omar Boussaid et le Pr. Stefano Rizzi pour leurs précieux conseils et le temps qu'ils nous ont accordé.

Références

Bakillah, M., M. A. Mostafavi, et Y. Bédard (2006). A semantic similarity model for mapping between evolving geospatial data cubes. In R. Meersman, Z. Tari, et P. Herrero (Eds.), *On the Move to Meaningful Internet Systems 2006 : OTM 2006 Workshops*, Berlin, Heidelberg, pp. 1658–1669. Springer Berlin Heidelberg.

Bimonte, S., O. Boucelma, O. Machabert, et S. Sellami (2014). A new spatial olap approach for the analysis of volunteered geographic information. *Computers, Environment and Urban Systems 48*, 111 – 123.

Bimonte, S., E. Edoh-alove, H. Nazih, M.-A. Kang, et S. Rizzi (2013). Protolap : Rapid olap prototyping with on-demand data supply. In *Proceedings of the Sixteenth International Workshop on Data Warehousing and OLAP*, DOLAP '13, NY, USA, pp. 61–66. ACM.

Brabham, D. C. (2008). Crowdsourcing as a model for problem solving : An introduction and cases. *Convergence 14*(1), 75–90.

Chen, L., K. S. Soliman, E. Mao, et M. N. Frolick (2000). Measuring user satisfaction with data warehouses : an exploratory study. *Information & Management 37*(3), 103 – 110.

Corr, L. et J. Stagnitto (2011). *Agile Data Warehouse Design : Collaborative Dimensional Modeling, from Whiteboard to Star Schema*. DecisionOne Press.

Cravero, A. et S. Sepúlveda (2014). Multidimensional design paradigms for data warehouses : A systematic mapping study. *Journal of Software Engineering and Applications 7*, 53–61.

Driedger, S. M., A. Kothari, J. Morrison, M. Sawada, E. J. Crighton, et I. D. Graham (2007). Correction : Using participatory design to develop (public) health decision support systems through gis. *International Journal of Health Geographics 6*(1), 53.

Egyed, A. et P. Grunbacher (2004). Identifying requirements conflicts and cooperation : how quality attributes and automated traceability can help. *IEEE Software 21*(6), 50–58.

Ehn, P. (2008). Participation in design things. In *Proceedings of the Tenth Anniversary Conference on Participatory Design 2008*, PDC '08, Indianapolis, IN, USA, pp. 92–101. Indiana University.

Gavish, B. et J. H. Gerdes (1997). Voting mechanisms and their implications in a gdss environment. *Annals of Operations Research 71*(0), 41–74.

Golfarelli, M. et S. Rizzi (2011). Data warehouse testing : A prototype-based methodology. *Information and Software Technology 53*(11), 1183 – 1198. AMOST 2010.

Kimball, R., M. Ross, J. Mundy, et W. Thornthwaite (2015). *The kimball group reader : Relentlessly practical tools for data warehousing and business intelligence remastered collection*. John Wiley & Sons.

Levrel, H., B. Fontaine, P.-Y. Henry, F. Jiguet, R. Julliard, C. Kerbiriou, et D. Couvet (2010). Balancing state and volunteer investment in biodiversity monitoring for the implementation of cbd indicators : A french example. *Ecological Economics 69*(7), 1580 – 1586. Special Section : Ecosystem Services Valuation in China.

Nabli, A., J. Feki, et F. Gargouri (2005). Automatic construction of multidimensional schema from olap requirements. In *The 3rd ACS/IEEE International Conference onComputer Systems and Applications, 2005.*, pp. 28–.

Naeem, M. A., S. Ullah, et I. S. Bajwa (2012). Interacting with data warehouse by using a natural language interface. In G. Bouma, A. Ittoo, E. Métais, et H. Wortmann (Eds.), *Natural Language Processing and Information Systems*, Berlin, Heidelberg, pp. 372–377. Springer Berlin Heidelberg.

Nuseibeh, B. et S. Easterbrook (2000). Requirements engineering : A roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, ICSE '00, New York, NY, USA, pp. 35–46. ACM.

Pohl, K. (2010). *Requirements engineering : fundamentals, principles, and techniques*. Springer Publishing Company, Incorporated.

Romero, O. et A. Abelló (2009). A survey of multidimensional modeling methodologies. *International Journal of Data Warehousing and Mining 5*(2), 1.

Stefanovic, N., J. Han, et K. Koperski (2000). Object-based selective materialization for efficient implementation of spatial data cubes. *IEEE Transactions on Knowledge and Data Engineering 12*(6), 938–958.

Sui, D., S. Elwood, et M. Goodchild (2012). *Crowdsourcing geo knowledge : volunteered geo information (VGI) in theory and practice*. Springer Science & Business Media.

Zaraté, P. (2013). *Tools for collaborative decision-making*. John Wiley & Sons.

Summary

In the context of voluntary geographic information (VGI), volunteers are not involved in the decision-making process. In addition, VGI systems do not offer powerful tools for conducting temporal analyzes. In this article, we propose to use the decision-making information systems to analyze VGI data, and we propose the definition of a new data warehouse design methodology, which allows the involvement of volunteers in the definition of analytical needs on VGI data. Our proposals have been tested on a real case study about biodiversity.