Handling spatial vagueness issues in SOLAP datacubes by introducing a risk-aware approach in their design

Djogbénuyè Akpé Edoh-Alove

N° d'ordre : D. U : 2566
E D S P I C : 696

UNIVERSITE CLERMONT AUVERGNE

ECOLE DOCTORALE
SCIENCES POUR L'INGENIEUR DE CLERMONT-FERRAND

T h è s e

Présentée par

DJOGBENUYE AKPE EDOH-ALOVE

Ingénieur, Master

pour obtenir le grade de

DOCTEUR D'UNIVERSITÉ

SPECIALITE : INFORMATIQUE

Proposition d'une nouvelle méthode de conception de cubes SOLAP exploitant des données spatiales vagues

Soutenue publiquement le 10 avril 2015 devant le jury :

M. ou Mme	Farouk Toumani	Président
	Christophe Claramunt	Rapporteur
	Omar Boussaid	Rapporteur
	Jacynthe Pouliot	Examinatrice
	Sandro Bimonte	Conseiller
	François Pinet	Co-Directeur de thèse
	Yvan Bédard	Co-Directeur de thèse

Handling spatial vagueness issues in SOLAP datacubes by introducing a risk-aware approach in their design

Thèse en cotutelle
Doctorat en sciences géomatiques

Djogbénuyè Akpé Edoh-Alove

Université Laval
Québec, Canada
Philosophiæ Doctor (Ph. D.)

et

Université Blaise-Pascal
Clermont-Ferrand, France
Docteur

Résumé court

Les systèmes Spatial On-Line Analytical Processing (SOLAP) permettent de prendre en charge l'analyse multidimensionnelle en ligne d'un très grand volume de données ayant une référence spatiale. Dans ces systèmes, le vague spatial n'est généralement pas pris en compte, ce qui peut être source d'erreurs dans les analyses et les interprétations des cubes de données SOLAP, effectuées par les utilisateurs finaux. Bien qu'il existe des modèles d'objets ad-hoc pour gérer le vague spatial, l'implantation de ces modèles dans les systèmes SOLAP est encore à l'état embryonnaire. En outre, l'introduction de tels modèles dans les systèmes SOLAP accroit la complexité de l'analyse au détriment de l'utilisabilité dans bon nombre de contextes applicatifs. Dans cette thèse nous nous proposons d'investiguer la piste d'une nouvelle approche visant un compromis approprié entre l'exactitude théorique de la réponse au vague spatial, la facilité d'implantation dans les systèmes SOLAP existants et l'utilisabilité des cubes de données fournis aux utilisateurs finaux.

Les objectifs de cette thèse sont donc de jeter les bases d'une approche de conception de cube SOLAP où la gestion du vague est remplacée par la gestion des risques de mauvaises interprétations induits, d'en définir les principes d'une implantation pratique et d'en démontrer les avantages.

En résultats aux travaux menés, une approche de conception de cubes SOLAP où le risque de mauvaise interprétation est considéré et géré de manière itérative et en adéquation avec les sensibilités des utilisateurs finaux quant aux risques potentiels identifiés a été proposée; des outils formels à savoir un profil UML adapté, des fonctions de modification de schémas multidimensionnels pour construire les cubes souhaités, et un processus formel guidant de telles transformations de schémas ont été présentés; la vérification de la faisabilité de notre approche dans un cadre purement informatique avec la mise en œuvre de l'approche dans un outil CASE (Computed Aided Software Engineering) a aussi été présentée. Pour finir, nous avons pu valider le fait que l'approche fournisse non seulement des cubes aussi compréhensibles et donc utilisables que les cubes classiques, mais aussi des cubes où le vague n'est plus laissé de côté, sans aucun effort pour atténuer ses impacts sur les analyses et les prises de décision des utilisateurs finaux.

Mots clés : Cubes de données spatiales, SOLAP, vague spatial, approche de conception, risque d'usage

Abstract

SOLAP (Spatial On-Line Analytical Processing) systems support the online multi-dimensional analysis of a very large volume of data with a spatial reference. In these systems, the spatial vagueness is usually not taken into account, which can lead to errors in the SOLAP datacubes analyzes and interpretations end-users make. Although there are ad-hoc models of vague objects to manage the spatial vagueness, the implementation of these models in SOLAP systems is still in an embryonal state. In addition, the introduction of such models in SOLAP systems increases the complexity of the analysis at the expense of usability in many application contexts. In this thesis we propose to investigate the trail of a new approach that makes an appropriate compromise between the theoretical accuracy of the response to the spatial vagueness, the ease of implementation in existing SOLAP systems and the usability of datacubes provided to end users.

The objectives of this thesis are to lay the foundations of a SOLAP datacube design approach where spatial vagueness management in itself is replaced by the management of risks of misinterpretations induced by the vagueness, to define the principles of a practical implementation of the approach and to demonstrate its benefits.

The results of this thesis consist of a SOLAP datacube design approach where the risks of misinterpretation are considered and managed in an iterative manner and in line with the end users tolerance levels regarding those risks; formal tools namely a suitable UML (Unified Modeling Language) profile, multidimensional schemas transformation functions to help tailored the datacubes to end-users tolerance levels, and a formal process guiding such schemas transformation; verifying the feasibility of our approach in a computing context with the implementation of the approach in a CASE (Computed Aided Software Engineering) tool. Finally, we were able to validate that the approach provides SOLAP datacubes that are not only as comprehensible and thus usable as conventional datacubes but also datacubes where the spatial vagueness is not left out, with no effort to mitigate its impacts on analysis and decision making for end users.

Keywords: Spatial datacubes, SOLAP, spatial vagueness, design approach, risk of usage

Résumé long

Les technologies Spatial-OLAP (SOLAP), dédiées à l'analyse multidimensionnelle de grands volumes de données (spatiales), ne tiennent généralement pas compte de l'incertitude sur les données spatiales, notamment le vague spatial. En effet dans ces systèmes, les objets spatiaux sont considérés comme ayant des limites et des positions bien définies et sont donc représentés par des géométries « crisp » (point, ligne polygone). Cependant, ces objets spatiaux représentent dans la majorité des cas des phénomènes géographiques dont les limites sont larges ou dont la position ne peut être connue avec précision (vague spatial). C'est l'exemple des zones d'inondation où les frontières sont comprises entre les limites des plans d'eau et les limites maximales enregistrées lors des inondations; c'est également le cas lorsque les géométries d'objets spatiaux, par exemple des parcelles agricoles, sont prises de diverses sources et intégrées pour obtenir des géométries uniques. L'écart entre cette représentation « crisp » et le vague de ces objets est source d'erreur dans les analyses et les prises de décisions dans tous les systèmes, à plus forte raison dans les systèmes SOLAP dont les utilisateurs sont avant tout des décideurs, rarement au fait de l'incertitude pouvant être présente sur les données analysées. Bien que de nombreux travaux de recherche proposent des modèles d'objets vagues pour mieux représenter les données qui s'y prêtent, l'intégration de ces nouveaux modèles dans les systèmes SOLAP reste encore à mettre en œuvre de façon performante dans la pratique. Les bases de données, serveurs SOLAP, outils Extract Transform and Load (ETL) et clients SOLAP classiques sont encore à gérer des géométries crisp et ils le font très bien. Aussi, analyser de la donnée multidimensionnelle en plus des métadonnées introduites par l'exploitation des nouveaux modèles peut-être trop complexe ou exigeant pour bon nombre de décideurs, dans la majorité des contextes applicatifs et pour certains besoins. Il se pose alors un défi d'intérêt pour les utilisateurs des systèmes SOLAP : Comment exploiter des cubes de données SOLAP classiques (implémentées avec les outils classiques donc) en tenant compte du vague spatial et ce de manière classique, simple et fiable?

Pour aider à relever ce défi, en d'autres mots, à réduire les conséquences du vague spatial sur l'exactitude des requêtes d'analyse SOLAP dans les systèmes classiques, nous proposons une nouvelle approche pour gérer le vague spatial dans les systèmes SOLAP. Cette approche est élaborée dans une vision symbiotique faisant un compromis entre l'exactitude théorique en termes de gestion du vague spatial, la fiabilité en terme de prise en compte du vague spatial et l'utilisabilité des cubes de données SOLAP fournis aux décideurs. Cette thèse propose donc l'introduction de la gestion des risques de mauvaises interprétations induits par le vague spatial dans la conception même des cubes attendus par les utilisateurs. En d'autres termes, il s'agit de concevoir et de transformer les schémas des cubes de données SOLAP suivant les niveaux de tolérance des utilisateurs finaux par rapport aux risques de mauvaises interprétations. Nous supposons que cette gestion de

risques en lieu et place de la gestion du vague (présent dans les sources) en lui-même permettrait de fournir aux utilisateurs finaux des cubes de données SOLAP suffisamment fiables et faciles d'utilisation pour leurs prises de décisions sans pour autant être la réponse parfaite au vague spatial. Pour ce faire, les bases d'une approche de conception orienté utilisateurs, orienté sources et consciente des risques de mauvaises interprétations relatives aux cubes SOLAP ont été définies: le risque de mauvaise interprétation a été défini et catégorisé et les principales étapes de conception consciente du risque ont été identifiées et décrites. Une emphase a été mise sur l'échelle de tolérance permettant de recueillir les sensibilités des utilisateurs finaux par rapport aux risques identifiés, ainsi que les différentes stratégies et actions de réduction de risque possibles. Les actions sont à appliquer sur les schémas en cours de conception des cubes SOLAP attendus en fonction des niveaux de tolérances. Par la suite, une méthode de conception mettant en œuvre l'approche a été élaborée. Cette méthode prend en entrée les besoins en analyse des utilisateurs, les données sources et fournit des cubes SOLAP adaptés aux paramètres de tolérance exprimés par les utilisateurs finaux au cours de la conception. Cette méthode a été implémentée dans un système de prototypage rapide et testée, dans le cadre de la validation de l'approche, sur des données d'épandage de boue (France).

Cette thèse a apporté des réponses intéressantes à nos questions exprimées précédemment. En effet, grâce aux différents travaux menés, nous avons pu ressortir comme premier résultat une approche de conception de cubes SOLAP où le risque de mauvaise interprétation est considéré et géré de manière itérative et en adéquation avec les sensibilités des utilisateurs finaux quant aux risques potentiels identifiés. Ensuite, cette thèse a permis de proposer des outils formels aidant à cette conception consciente du risque : un profil UML adapté, des fonctions de modification de schémas multidimensionnels pour construire les cubes souhaités par les utilisateurs au fur et à mesure de l'expression de leurs tolérances aux risques, et enfin un processus formel guidant de telles transformations de schémas. Un autre des résultats intéressants obtenus est la vérification de la faisabilité de notre approche dans un cadre purement informatique avec la mise en œuvre de l'approche dans un outil CASE développé au sein de notre équipe de recherche. Ce résultat nous indique notamment que non seulement une telle approche est possible mais qu'elle est également implantable à moindre coût (temps et complexité) avec les outils existants. Pour finir, nous avons également pu valider le fait que l'approche fournisse non seulement des cubes aussi compréhensibles (nombre similaires de classes, de hiérarchies spatiales et de dimensions spatiales ainsi que nombre inférieur de hiérarchies multiples et de mesures par fait pour les cubes testés) et donc utilisables que les cubes classiques, mais aussi des cubes où le vague n'est plus laissé de côté, sans aucun effort pour atténuer ses impacts sur les analyses et les prises de décision des utilisateurs finaux.

Table des matières

Liste des tableaux

Liste des figures

A Dieu mon Créateur,
A ma famille bien-aimée,
A ma famille Choux du Québec,
A mes amis et à mon futur mari

Remerciements

Il était une fois une jeune fille togolaise qui rêvait de faire une thèse en Géomatique. L'histoire ne nous dit pas comment elle est tombée dans la Géomatique mais ce n'est pas le plus important. Cette jeune fille, c'est moi bien entendu, et je me ferais un plaisir de vous raconter cette histoire-là à une autre occasion. Donc après des années d'études inoubliables à l'École Hassania des Travaux Publiques (EHTP) au Maroc, je me suis retrouvée dans un avion pour le Canada afin d'y effectuer une maîtrise avec l'aide financier de l'Agence Canadienne pour le Développement International (ACDI) en 2009. J'avoue que l'idée à l'époque était d'enchaîner directement sur une thèse comme je le rêvais mais que nenni, à l'Université Laval (UL) c'est d'abord la mini-thèse (mémoire de maîtrise) puis la grande thèse (thèse de doctorat). Peu importe, je n'allais pas me décourager pour si peu. Deux années de dures labeurs en bonne compagnie plus tard, j'ai eu l'opportunité de décrocher mon sujet de thèse : Eurêka!!! Cette thèse non seulement me semblait passionnante, mais en plus elle présentait une particularité plutôt rare : la cotutelle franco-québécoise. Pour ceux qui ne sont pas familiers avec le concept je vous le fais en deux mots : une thèse en cotutelle, c'est l'opportunité d'être rattaché à deux laboratoires différents, dans deux pays différents, de travailler avec deux directeurs de recherche, dans deux systèmes différents et le must, de décrocher non pas un mais deux diplômes de doctorat (dans mon cas un diplôme français et un canadien). Je ne pouvais pas rêver mieux. Je n'avais même pas encore fait mon dépôt initial de mémoire de maîtrise que me voilà embarquée dans une nouvelle aventure. Je me suis vite rendu compte cependant que cette cotutelle représentait tout un défi. En effet, non seulement ma thèse a l'obligation de répondre aux exigences de chacune des universités (UL et Université Blaise Pascal de Clermont-Ferrand) et de chaque département/école doctorale impliqués, mais il me faut également composer avec deux mondes différents mais complémentaires que sont le monde de la Géomatique et celui de l'Informatique. Il me fallait donc par exemple valider des crédits de cours obligatoires et un examen de doctorat selon les critères du département des Sciences Géomatiques de l'UL la première année et valider des modules obligatoires du côté de l'Université Blaise Pascal les deux premières années; aussi mes contributions se doivent d'être compréhensibles et pertinentes à la fois pour les géomaticiens et pour les informaticiens. Je dois avouer que tout cela rajoute une grosse difficulté et une dose de travail supplémentaire. N'eut été l'aide de mes directeurs de recherche, tant ceux de la maîtrise que de la thèse, de mes amis, de mes collègues et de ma famille, je n'aurais surement pas pu voir le bout du tunnel. Je tiens donc à remercier du fond de mon cœur chaque personne qui aurait contribué de près ou de loin au succès de ce projet de vie et à mon épanouissement tant sur le plan personnel que professionnel.

En particulier, je remercie mes directeurs de recherche Yvan Bédard et François Pinet ainsi que mon conseiller Sandro Bimonte: vous m'avez encouragé, soutenu, motivé et poussé à donner le meilleur de moi-même; cette énergie donnée se retrouve dans ces pages mais également dans mes veines pour toujours. Mes sincères remerciements vont également à Jacynthe Pouliot, Christophe Claramunt, Omar

Boussaid, Farouk Toumani, examinateurs et membres du jury qui m'ont fait l'honneur d'évaluer ma thèse et de contribuer à son achèvement dans les meilleures conditions.

Je remercie ma famille et mes amis dispersés aux quatre coins du monde d'avoir cru en moi, d'avoir toujours trouvé les mots pour m'encourager, d'avoir donné sans compter tous ces câlins qui me réconfortaient et surtout d'avoir fait preuve de patience quand je ne vivais et ne respirais que pour ma thèse. Vous êtes des amours et je suis bénie de vous avoir dans ma vie. Je comptais ne pas vous citer de peur d'oublier un seul prénom, mais après tout, je trouve que manqueriez à ces remerciements. Alors à toutes les personnes dont les noms vont suivre, je vous aime de tout mon cœur et merci d'être là pour moi: ma maman chérie Élisabeth, ma sœur Édith, mon grand frère Mawunyo Eli, mon papa Émile, mes amies de longue date, Dédé et Atia, mes amis rencontrés au Maroc Hind, Lamiae, Mouna et Eric, mes sœurs et frères choux rencontrés au Québec Ramatou, Akin, Nadège, Serge, Maréva, Priscille, Rokya, Patricia, Pertine, Samuel, Adrienne, Déborah, Marie-Louise et Rose, mes amis rencontrés à Clermont-Ferrand Patrick, Nelly, Armel et Joce et enfin Sarah ma nouvelle collègue et passagère de covoiturage qui n'hésite jamais à prendre le volant pour me laisser travailler ou me reposer un peu plus aux sorties de mes nuits d'études.

Je remercie mes collègues et amis, ceux du CRG (Centre de Recherche en Géomatique), Alborz, Mojgan, Eve, Tania, Hedia, Naouraz, Chen, Danielle, et ceux d'Irstea (Institut de Recherche en Sciences et Technologies pour l'Environnement et l'Agriculture) Mehdi, Pawan, Fabien, Chloé, Eva et Vincent pour leur écoute, leur disponibilité et leur aide précieuse à un moment ou à un autre durant ces trois ans de thèse.

Toute ma gratitude va également à Frédéric Hubert, Thierry Badard, mes directeurs de maîtrise, Jacynthe Pouliot et Nicholas Chrismann, deux professeurs qui ont influencé ma vision de la géomatique, Thomas et Guillaume deux collègues et amis qui ont rendu mes journées au CRG beaucoup plus agréables, toute la communauté des étudiants catholiques de l'Université Laval avec à sa tête le père Jean Abud, les belles personnes que j'ai rencontrées à travers le CCO à savoir Talitha, Céliane et Émilie.

Je remercie enfin Dieu pour tout l'Amour dont il comble mon cœur car sans tout cet amour, je n'aurais pu rien accomplir.

Musique écoutée pendant la rédaction : Sarah Bareilles (avec une préférence pour Brave et Love song), Sam Smith, Daley, Stromae, compil de Jazz et de musiques religieuses.

Avant-Propos

Ce projet de thèse a été réalisé dans le cadre d'une cotutelle Géomatique/Informatique entre l'Université Laval au Québec (Canada) et l'Université Blaise Pascal à Clermont-Ferrand (France). La première année s'est déroulée dans le Centre de Recherche en Géomatique de l'Université Laval, puis les deux années restantes ont été passées à l'Institut de Recherche en Sciences et Technologies pour l'Environnement et l'Agriculture (Irstea), site de Clermont-Ferrand. Il a été mené sous la direction du Pr. Yvan Bédard (Université Laval) et du Dr. François Pinet (Université Blaise-Pascal).

Ce document est présenté pour rencontrer les critères de réalisation des deux universités de cotutelle. Il est écrit en anglais et est articulé autour de trois publications principales faisant état des résultats de nos travaux. Les trois publications ont été soumises à différentes revues avec comité de lecture et ont, tout comme cette thèse, été rédigées par l'étudiante (Elodie Edoh-Alové) qui en est donc l'auteure principale ainsi que la responsable de leurs soumissions. Le conseiller Sandro Bimonte, ainsi que Yvan Bédard et François Pinet, directeurs de ce projet de thèse, sont les co-auteurs des publications. Elles ont toutes les trois été ajustées afin d'obtenir un document facile à lire et cohérent.

Le **premier article** constitue le Chapitre 3 de ce document. La section sur l'état de l'art a été enrichie et nous avons adapté le style et la mise en forme pour le rendre conforme au présent document. L'article a été accepté pour publication en Juillet 2015 dans la revue International Journal of Agricultural and Environmental Information Systems (IJAEIS) sous la référence :

Edoh-Alove E., Bimonte, S., Pinet F., & Bédard Y. (2015). "New Design Approach to Handle Spatial Vagueness in Spatial OLAP Datacubes: Application to Agri-environmental Data", IJAEIS 6(3).

Le **second article** quant à lui se retrouve en grande majorité dans le Chapitre 4 du présent document. L'article original a été enrichi avec une section plus complète sur notre méthode RADSOLAP et la mise en forme et le style ont été adaptés pour le présent document. Il a été soumis et accepté pour publication dans la revue International Journal of Data Warehousing and Mining (IJDWM) sous la référence :

Edoh-Alove E., Bimonte S., & Pinet F. (to be published) "An UML profile and SOLAP datacubes multidimensional schemas transformation process for datacubes risk-aware design", IJDWM.

Enfin le **troisième article** est repris dans le Chapitre 5 de ce document. La version courte de cet article a été présentée et publiée dans les actes de la conférence internationale ICCSA 2014 sous la référence :

Edoh-Alove, E., Bimonte, S., & Bédard, Y. (2014). A New Design Method for Managing Spatial Vagueness in Classical Relational Spatial OLAP Architectures. In Computational Science and Its Applications–ICCSA 2014 (pp. 774-786). Springer International Publishing.

A la suite de quoi, cet article a été retenu puis accepté dans sa version étendue pour publication dans une édition spéciale de la revue International Journal of Business Intelligence and Data Mining (IJBIDM), sous la référence :

Edoh-Alove, E., Bimonte S., Bédard, Y. & Pinet, F. (in Press). "A hybrid risk-aware design method for spatial datacubes handling spatial vague data: implementation and validation", IJBIDM.

Cette version étendue a été donc ajustée pour le Chapitre 5. Nous avons notamment enlevé la section sur la méthode RADSOLAP, section qui rappelons-le se trouve dans le Chapitre 4, et rajouté les diagrammes de comparaison pour les critères d'utilisabilité des cubes SOLAP. Nous avons également adapté le style et la mise en forme tout comme pour les autres articles.

Chapter 1: Introduction

1.1 Research context

1.1.1 Spatial vagueness: an often neglected spatial data uncertainty

The spatial data is usually the representation of an observed reality (Worboys 1998), more specifically a phenomenon that can be located in the real world, whether it is natural (lakes, forests, mountains etc.) or defined by humans activities (planned roads, buildings, electoral boundaries, census areas etc.). It is often subject to uncertainty which is caused in general, by the limitations of the modeling process (models too simplified, omission of details, definition of the observed reality itself as illustrated in the previously etc.), the measurements tools and/or the observers themselves (Bédard 1986). The uncertainty affects the spatial data conception, position, shape, attributes, and temporal accuracy (Devillers and Jeansoulin 2006) and exposes spatial data users to faulty analysis and data interpretation afterwards. Indeed, for instance, an uncertainty on the conception of an object (what do we call swamp? Where does the swamp stop and where does the pond start), will in turn creates a thematic inaccuracy on the spatial data (the entity is not well classified), and the entity attributes, even if they are well measured, might be inaccurate because they are not the right attributes describing that entity. Using such data, analysts can have error in their comparison and statistics with a more or less great damage.

Since few decades, many researchers have tackled the spatial data uncertainty issues in order to describe, minimize and/or report it to potential spatial data users (Bédard 1986, Beard 1989, Burrough and Frank 1996, Lagacherie, Andrieux et al. 1996, Dilo 2006). They have worked out different techniques and models (error models, measurements compensation, confusion matrix etc.) to manage the spatial data uncertainty in the various Geomatics application fields (Remote sensing and image interpretation, topography, Geographic Information Systems (GIS) etc.)

Spatial vagueness, in particular is an uncertainty presents on the majority of spatial data related to natural environment (Burrough and Frank 1996). In fact, the majority of natural phenomena are characterized by:

- A difficulty to find limits that correspond to physical discontinuity (where does the lake stop and where does the ground start?). Their shapes (including their boundaries) are not clear (vague shapes).

- A difficulty to position them precisely due to a lack of knowledge about their location (vague location).

However, as any other geographic object, such natural phenomena are most of the time represented as spatial objects with definite boundaries separating their interior from their exterior on maps, in spatial data bases and GIS in general. It is a crisp representation (e.g. point, line, and polygon geometry types), chosen for the sake of simplification and manipulation of spatial objects in information systems. This choice of simplification creates the spatial vagueness on the spatial data presented to data consumers.

Now, GIS are democratized via web applications (Devillers, Stein et al. 2010) and a huge amount of spatial data are available in geospatial infrastructures on the internet. Anybody can access spatial data and use them. Users, especially those that are not spatial data experts are, very often, not aware of the possible spatial data uncertainty, in particular spatial vagueness; therefore they often cannot take it into account in their analysis and decisions.

Not considering the spatial vagueness in decision-support systems can represent a more or less great danger for the end-users. Let us consider the example of the lakes that have vague shape. For a user that needs to simply locate a lake for a navigation purpose, having a map that represents the lakes as crisp polygons is sufficient and does not present a real danger. For another user that needs to know the surface of lakes in a region in order to plan lake's inhabitant (animals and plants) safeguarding, the same data will lead him to an improper estimated surfaces; he would take his decisions based on non-reliable information with a money loss and environment damage as ultimate result.

1.1.2 A need for spatio-multidimensional analysis

Over the years, companies and public organisms in different fields have accumulated a huge amount of data in their transactional (spatial) databases. Those data hold knowledge and information useful to the decisional process. For example, one can extract tendencies from the data by doing temporal and thematic synthesis, spatial comparisons, etc.(Bédard and Han 2009). In this Web 2.0 era, there is a certain need for tools that support efficiently such data interrogation and exploration in an easy, quick and reliable manner.

For that purpose, Spatial OLAP[1] (SOLAP) systems have emerged in the past decade. SOLAP systems are *visual platforms that allow easy, rapid and interactive exploration of huge volume of data by simple clicks on pivot table, histograms or maps that present queries results to the users* (Bédard, Rivest et al. 2006). To do so, they are most of the time based on a multidimensional approach to structure an organization data (Proulx and Bédard 2004). This approach introduces new concepts ("dimension", "measure, "fact" etc.), unknown to the transactional systems, that makes the data structure more close to the mental representation of the data users have (Codd, Codd et al. 1993).

[1] On-Line Analytical Processing

Indeed, in the multidimensional approach, historic data on agricultural activities in France, for example, are structured as such (Bédard, Merrett et al. 2001):

- The analysis themes are represented as dimensions *Time*, *Operations* and *Location.* They contain members (analysis objects) organized into hierarchies according to their granularity levels (e.g. *Farming plots* grouped into *Farms*, grouped in turn into administrative *Departments*, grouped in turn into administrative *Regions,* grouped in turn into *France*).

- The analyzed values are represented by the measure, *Quantity of Energy Used*. A measure in general is a dependent variable that reports on the situation in regards with the analysis themes.

With that structure, a multidimensional query can be simply defined by saying "I want to know the *Quantity of Energy Used* (a measure) for the operation *Labour* relatively to the farming plot FP1 during the year *2000* (dimensions members)".

Note that combination of dimensions and measures are called facts and are stored in a table and queried by the users. A fact is for example "the quantity of energy used by the operation *Labour* relatively to the farming plot *FP1* during the year *2000* is 200kJ".

Most of the SOLAP systems store the facts in datacubes (hypercubes actually) which is basically the physical implementation of the spatio-multidimensional model.

Nowadays, SOLAP systems find more and more applications in many fields including health, finance, transport, agriculture, environment, etc. Their users are first and foremost decision-makers, expert analysts or application field professionals (Guimond 2005) that are in majority unaware of the spatial data uncertainty issues, in particular the spatial vagueness. Those users rely on the analyses results to make important strategic decisions whose consequences (positive or negative) can be far reaching. Therefore, they cannot afford missed problems, inexact comparisons between regions or faulty trend analysis in general.

1.2 Problem definition

We believe that it is imperative to deal with spatial vagueness issues in SOLAP systems, somehow. The spatial vagueness is at the source of diverse spatial data quality issues in a SOLAP datacube. First of all, a quality issue present on the sources spatial data (and at the finest level of the datacube) has an impact on the whole datacube quality. For instance, if the member Rhône is absent in the spatial dimension, there is an absence of measures and facts for that member at least at the corresponding aggregation level. Since the spatial data quality is correlated to the spatial data uncertainty, having uncertainty in the sources leads to spatial data quality issues on the SOLAP datacube. More specifically, spatial vagueness on members or facts attributes can brings inaccuracy (incompleteness, attribute inaccuracy, spatial

inaccuracy, etc.) not only on the geometries themselves (such as cities boundaries, fire regions or agency positions contained by members, members' attributes and facts attributes), but also on the other members of the spatial hierarchy and the aggregation results (non-geometric and geometric facts attributes). Let us take the example of crop areas with vague shapes. The measures calculated based on the crop areas, whether it is a spatial measure such as the surface of the crop area, or non-spatial measure such as the quantity of energy spent on the crop area are subject to spatial or attribute inaccuracy respectively. Also, the spatial intersections and other predicates introduce uncertainty on the aggregates calculated based on those crop-areas geometries.

To cope with spatial vagueness, very recently, some researchers (Jadidi, Mostafavi et al. 2012, Siqueira, Ciferri et al. 2014) have advocated the use of spatial vague objects models (fuzzy models, exact models, rough models etc.) to represent vague phenomena (Zadeh 1965, Cohn and Gotts 1996, Lagacherie, Andrieux et al. 1996, Schneider 1999, Dilo 2006, Bejaoui 2009, Pauly and Schneider 2010). Vague objects models are designed to be more truthful to the reality of such phenomena: they allow for instance the representation of broad shapes and vague locations by means of:

- Fuzzy sets (fuzzy models): *an object is a set of individuals+ their membership degrees expressing the possibility that the individual belongs to the set*;

- Egg-yolk concepts (exact models): *an object is a combination of a core (certain part) and a dubiety (uncertain part).*

Therefore the distortion between the real object and its description (spatial vagueness) can be reduced. More specifically, they have proposed new spatio-multidimensional models and SOLAP operators that support the spatial vague objects models. In the following paragraphs we introduce those proposals to draw out some of their limits in order to justify the problem addressed in this thesis (For more details, refer to Chapter 2).

Jadidi, Mostafavi et al. (2012) introduce the fuzzy logic (Dilo 2006) into spatial datacubes. They define the fundamental concepts needed to model and exploit objects with fuzzy semantic, vague shapes and fuzzy temporality in spatio-multidimensional based geo-decisional systems. The proposals are applied to coastal erosion risk areas. They include membership functions to define the objects (fuzzy sets), formal definition of the fuzzy spatial datacube elements (fuzzy spatial dimension, spatial fact, spatial measure etc.) and some new fuzzy spatial aggregation operators (fuzzy union, intersection, overlay etc.).The implementation of the fuzzy spatial datacube is identical to classic ones; nevertheless, the measures should be calculated with the fuzzy operators instead of the classic ones. Note that the approach is usually dedicated to raster data types. The implementation of the defined operators is yet to be done in classical SOLAP tools (and in traditional DBMSs) and also the computation of the membership functions requires specific tools that are

not available. On top of that, the solution calls for a case-by-case definition of the membership functions, a task that is not always easy (Dilo 2006).

On their part, Siqueira, Ciferri et al. (2014) introduce the exact models (Bejaoui 2009, Pauly and Schneider 2010) into spatial datacubes. They define the VSCube model, which is a spatio-multidimensional model that manages vague objects (vector data types). Their proposals include a formal definition of the VSCube elements (vague spatial attribute, vague spatial level, vague spatial measure etc.) as well as specific techniques for querying (vague window query, vague SOLAP operators) and storing the vague SOLAP datacubes. Regarding the storing, they propose to store separately the core and the dubieties parts (plus values in the interval]0,1[expressing the uncertainty degrees) of a vague object. Just as the previous approach, the implementation of the new definitions and techniques is yet in classical SOLAP tools is yet to be proposed and effective.

Even though vague objects models are in theory the most accurate approach to represent spatial vague objects, introducing such models in SOLAP datacubes jeopardize the datacubes usability and the ability to easily implement them in current classical tools.

Indeed, the introduction of complex (in opposition to the simplicity of the classical point line polygon) spatial objects models in datacubes always comes with more data to visualize, and to digest (complex geometry + uncertainty values). Also, it can be too much information for end-users regarding their context of use. In fact, while in some cases, they absolutely need the most accurate solution to take into account the spatial vagueness (it is for example the case of coastal erosion risk assessment), in other cases, the complexity of the solution can just be too much for the intended usage. For example, an end-user who wants to exploit the same coastal erosion risks zone in a datacube that will help him analyze the investments regarding such regions over the years, it is sufficient to show the approximate regions as simple polygons (classic spatial level) with the investment values (the measure). Therefore, in this case, if provided with a datacube integrating the fuzzy logic to handle the zones vagueness, he will not only have to make a good effort to comprehend the data but also it will be an useless effort since his intended use does not need this accurate but complex representation. Hence the usability of this particular solution is compromised for this end-user.

On the other hand, we recall that using just the simple crisp geographic data types without any consideration for the spatial vagueness leads to a non-reliable SOLAP datacube putting end-users under risks of misinterpretations. However till now, when it comes to vector data, classical SOLAP server and client as well as spatial DBMS are designed for the manipulation of crisp geographic data types that are the point, line and polygon. To benefit from their full potential with no delay and no extra effort from datacubes producers and to guarantee an easy implementation of produced SOLAP datacubes, it is

required to **stick with the crisp geographic data types (point, line, and polygon) to represent all spatial objects, vague or not.**

To stick with crisp geographic data types and still take spatial vagueness into account in a simple way, we find judicious to replace the spatial vagueness management itself by the management of risks of data misinterpretation induced by the spatial vagueness.

Indeed, for years now the research regarding projects risk management (Del Cano and de la Cruz 2002) in general, and software risk management in particular (Boehm 1991, Karolak and Karolak 1995) have established that an early concern with identifying, analyzing and controlling risk elements helps prevent providing users with unsatisfactory solutions (wrong functionalities, solutions with performance or reliability issues etc.). In the same line, it is increasingly recognized that spatial data risks of misuse management in general is required to avoid unexpected results and faulty trend analysis to spatial data consumers, regardless the data quality itself (Lévesque 2008, Gervais, Bédard et al. 2009, Gervais, Bédard et al. 2012, Grira, Bédard et al. 2013, Roy 2013). The reason is that nowadays, anybody can access spatial data through appropriate infrastructures on the internet, but not all consumers are aware of the data quality, let alone the usage for which the data was produced. Therefore they make inappropriate use of the data which lead them to unexpected results with consequences that can be far reaching. In this case (spatial vagueness management), we also want to prevent end-users from faulty trend analysis thus the idea to take this particular approach angle.

Doing a risk of misinterpretation management means that the risks need to be identified, assessed and reduced during the datacube design to provide end-users with appropriate, usable and more reliable SOLAP datacubes. To respect the usability condition, it is also required to **consider end-users tolerance to potential risks of data misinterpretation soon enough in the datacubes elements definition (production phase).**

Lévesque (2008), Grira, Bédard et al. (2013) and Roy (2013) have interesting contributions in regard with risks of misuse management. Their proposals comprise not only a risk of misuse management method and tools for risks identification and communication in SOLAP systems via pop-up alerts (Lévesque 2008), but also a collaborative platform to identify and analyze risks, relative to spatial data in general (Grira, Bédard et al. 2013), as well as an approach to better inform the spatial data consumers (Roy 2013). The contributions can be applied in the same system with the risks identification and documentation based on Lévesque (2008) formal tools that one can implement in Grira, Bédard et al. (2013) collaborative approach which will help involve end-users and collect risks management strategies. Then using Roy (2013) approach and Lévesque (2008) alerts system, the risks can be communicated to the end-users ultimately. When placing those contributions on a risks of spatial data misuse spectrum, it appears that there is a gap regarding the technical application of the risks management strategies on SOLAP datacubes during their

definition. Such proposal could be used to implement the collaborative approach results in order to insure more reliable datacubes and then any residual risk could be communicated afterwards. Regarding the definition of datacubes, there are many work covering datacubes (spatial or not) modeling process in the literature. Some researchers propose to draw the multidimensional structure from end-users needs in analysis (Böhnlein, Plaha et al. 2002, Lujan-Mora, Trujillo et al. 2006, Prat, Akoka et al. 2006); others offer approaches that extract multidimensional knowledge from available data-sources (Jensen, Holmgren et al. 2004, Malinowski and Zimányi 2008, Romero and Abelló 2008, Song, Piattini et al. 2008); finally, we can also find hybrid approaches (most used for SOLAP datacubes in particular) that based the multidimensional modeling not only on end-users needs in analysis but also on the available data sources (Guimond 2005, Mazón, Trujillo et al. 2007, Romero and Abelló 2010). Those approaches have been proven to be efficient and quick in furnishing datacubes models that fit end-users requirements. However, they do not take into account the risks and tolerance levels (expressed by end-users) on top of the end-users needs in analysis and the available resources. With all that said, it appears that **a hybrid SOLAP datacubes design process allowing the application of risks management rules, according to end-users sensibility, on SOLAP datacubes schemas is missing.** Such process would present the advantage of allowing a quick modeling and thus implementation of SOLAP datacubes where risks of misuse are not only detected and assessed but also reduced with the help of all parts involved in the project, especially end-users.

In summary, for all the reasons above, we think a new solution that helps designing SOLAP datacubes which are usable for any end-user and can be easily implemented in classical tools while taking spatial vagueness into account is required. This solution will come as an alternative for applications where it is not vital to represent the spatial vague objects with a perfect theoretical accuracy (fuzzy logic for example). In conclusion to our previous observations and reasoning, we believe that this alternative solution should be based on an approach making a symbiotic trade-off between the datacubes usability (is the datacube schema understandable and thus easily usable?), the ability to do an easy implementation of the datacubes in classical tools (Can the datacube be implemented without any extra effort on technical tools to manage complex vague models?) and the theoretical accuracy when dealing with spatial vagueness (To what extent the solution is correct theoretically?). As shown in the following Figure 1-1, right now, such solution is yet to be proposed in the literature. In fact, the existing solutions are whether spot on regarding the spatial vagueness management in theory (fuzzy set theory and other vague object models) (Jadidi, Mostafavi et al. 2012), or focused on the usability and ease of implementation (classical crisp representation case), or more on the theoretical accuracy side with a concern for the ease of implementation (exact models implementations in SOLAP systems) (Siqueira, Ciferri et al. 2014).

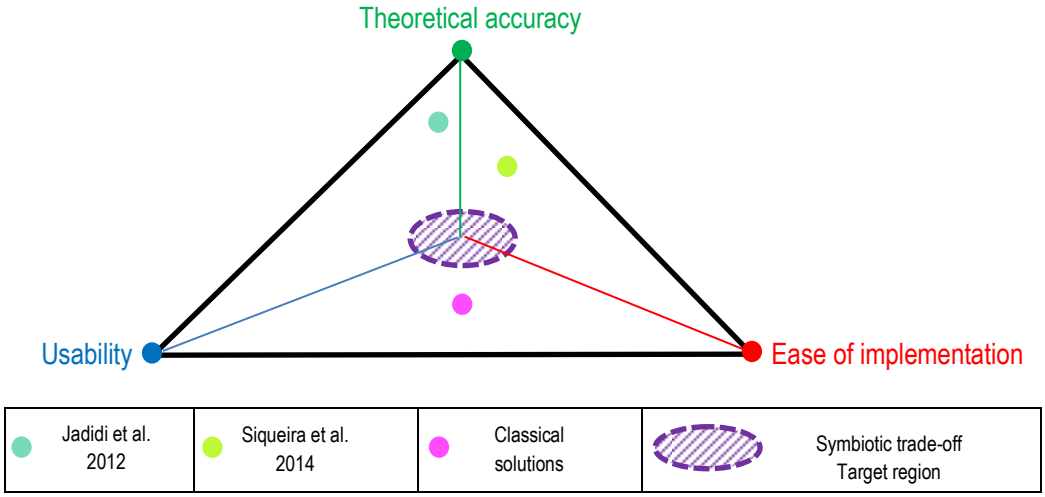


Figure 1-1 : Classification of the existing solutions regarding the three criteria: Usability, theoretical accuracy and ease of implementation

In the rest of this thesis, when talking about a symbiotic trade-off vision or approach, it specifically implies a trade-off between usability, easy implementation and theoretical accuracy regarding the spatial vagueness. Thereby, the general problem addressed in this thesis is:

There is no solution allowing the design of usable, more reliable and easily implementable SOLAP datacubes where spatial vagueness issues are handled in a symbiotic trade-off approach.

To make sure the problem, research questions and hypothesis are well understood, it is important to define the terms *usability*, *reliability* and *ease of implementation* as adopted in the context of this thesis.

- *Usability*: A usable datacube is a datacube which is understandable by end-users and fits their use (Serrano, Trujillo et al. 2007).

- *Reliability*: reliability here is attached with the notion of theoretical accuracy. A datacube is said, in this thesis, to be reliable when the spatial vagueness is considered regardless the extent to which the solution is accurate on representing vague objects.

- *Ease of implementation*: A datacube is easily implementable in classical systems and architectures (DBMS and SOLAP for example) when the implementation does not involve any new techniques and data types to store and query the data. By new, we mean that they are not already implemented in the classical systems.

1.3 Questions and hypothesis of research

The principal research question we answer is the following: **is it possible to produce SOLAP datacubes exploiting vague spatial objects that remain as usable, reliable and easily implementable as classical SOLAP datacubes?**

Other questions we answer in this research are:

- How to integrate the risk management method in a SOLAP datacubes design process?
- How to evaluate and integrate the tolerance levels of end-users, to the potential risks of misinterpretation, in the design process?
- How to implement such risk-aware SOLAP datacubes design process in a computing environment?
- Is it possible to benefit from existing design and development concepts and tools to support the new approach in practice?

The general hypothesis we formulate in this research is that **a design approach based on a symbiotic trade-off vision allows producing SOLAP datacubes, exploiting spatial vague objects, that remain as usable, reliable and easily implementable as classical SOLAP datacubes.**

The usability will be expressed in this thesis as the datacube schema understandability; the reliability will be translated into the fact that the spatial vagueness is considered or not and the ease of implementation will be expressed as the possibility of implementing the SOLAP datacubes in existing classical SOLAP architectures right away.

In the first place, such approach should allow collecting end-users needs in analysis and identify pertinent sources data in order to draw out a multidimensional knowledge (model plus aggregations). In the second place, it should allow the identification of the spatial vague objects present in the sources and to be used in the intended datacube. Then it should help identifying all potential risks of misinterpretation end-users incur as well as collecting their tolerance levels regarding the identified risks. The approach should also allow the identification of appropriate risks management actions (e.g. delete a spatial level that introduces an inacceptable risk in the datacube), actions which are used afterwards in the definition of the factual and multidimensional data (dimensions and hierarchies, measures, and aggregation formulas) that will compose the final SOLAP datacube.

1.4 Objectives

To verify our hypothesis, the general objective of this research consists of **proposing a SOLAP datacubes design method integrating a risk of misinterpretation management method to deal with spatial vagueness in a symbiotic trade-off approach.**

In this thesis project, it was not realistic to provide a generic and complete solution to all the possible case studies out there. We have preferred to provide the fundamentals of an efficient method, so as to verify that usable, more reliable and easily implementable SOLAP datacubes exploiting vague spatial objects can be provided by using the symbiotic trade-off vision. To reach this goal, we define three sub-objectives:

- **Sub-objective 1: To propose the fundamentals of a risk-aware SOLAP datacubes design approach**. This implies:

- To define how the spatial vagueness is taken into account in the symbiotic trade-off vision.

- To define the risk of misinterpretation, and the tolerance concept regarding SOLAP datacubes.

- To propose approaches to identify and manage the risks, and express the end-users tolerance to the identified risks.

- To integrate a risk management method with the classic SOLAP datacubes design process.

- **Sub-objective 2: To propose the principles of a practical implementation of the risk-aware approach.** In this thesis, because we wanted to achieve a spatial vagueness management in practice, it is important to provide also practical principles that will sustain the approach implementation by datacube producers. Here, we focus more on the informatics aspect of this research project. This sub-objective implies:

- To define an agile design process implementing such an approach.

- To define the principles of an UML-based multidimensional modeling to support the approach.

- To define and formalize the SOLAP datacubes schemas transformation process that allows tailoring the datacubes to end-users tolerance.

- **Sub-objective 3: To demonstrate the implementation feasibility of the proposed risk-aware approach.** This sub-objective is driven by the need for our Irstea research team to test the risk-aware approach in our own prototyping CASE system (called ProtOLAP) and by our third research question. The demonstration is done by proposing and testing an implementation of the new approach in the ProtOLAP tool.

- **Sub-objective 4: To demonstrate the benefits of the risk-aware approach.** This is done by evaluating and comparing the risk-aware approach results with a classical SOLAP datacubes design results. The three elements that were at the basis of the evaluation/comparison are: the ease of implementation, the reliability and the usability of the designed schemas. The SOLAP datacubes modeling was provided for a real sludge spreading case study relative to the SILLAGE (formerly SIGEMO) project (Soulignac, Barnabé et al. 2006).

1.5 Methodology

In this thesis, we adopt a methodology based on a hypothetico-deductive reasoning. After a literature review, a problem has been defined and our research question has been brought out. To answer this question, we have defined a research framework that consists of elaborating formal and technical tools to put our general hypothesis to test. Indeed, using those tools to design few datacubes, we are able to confirm the hypothesis or not. The methodology is divided 4 main phases:

Phase 1: Knowledge acquisition, literature review and problem definition

We have collected knowledge about spatial data uncertainty and SOLAP concepts, architectures and tools. We also did at first an in-depth literature review on spatial vagueness, SOLAP systems, and spatial vagueness management in SOLAP systems. This was necessary to understand the issues people are facing when trying to exploit spatial vague objects in any kind of information system, in particular in SOLAP datacubes. We have specifically studied the advantages and the limits of the existing spatial vagueness management approach (Jadidi, Mostafavi et al. 2012, Siqueira, Ciferri et al. 2014). Then we have also done a literature review on risks of misuse management in general and on multidimensional design methods. From there we have defined the problem addressed by this research project. Note that the literature review has been enriched and updated during the research with newly published work.

Phase 2: Proposal of the risk-aware design of SOLAP datacubes approach

In this phase, we worked towards meeting our first sub-objective. The very first step was trying to answer this question: “What happens when spatial vague objects are introduced in the SOLAP datacubes but spatial vagueness is not taken into account?” To do so, we have conducted a study of the impacts of having vague geometric attribute in spatial levels, on the SOLAP datacube at a conceptual level, e.g. what is the impact of the existence of a member with a vague shape on the hierarchy, the dimension, and the measure? Is the measure also uncertain? If yes, is it uncertain at all aggregated levels or just the level concerned? This study results helped us define afterwards the diversity of the misinterpretations end-users can face when using SOLAP datacubes exploiting spatial vague objects.

We have also worked out a simple geometry typology, using crisp geographic data types, to take into account the spatial vagueness in a simple way. This typology is based on the easy to use spatial vague

objects model named Qualitative Min-Max advocated by our colleague Bejaoui (2009). This first step is more detailed in the third chapter of this document.

The second step consisted of defining properly the risk of misinterpretation and classifying it in order to provide the basis for our approach. The risk of misinterpretation has been classified in two main categories, namely the intrinsic risks, induced by the data themselves, and the extrinsic risks which are related to the user context of use. Regarding the intrinsic risks, they are defined as risk of measure poor evaluation and classified in turn in three groups: risk of over evaluation, risk of under evaluation and non-significant risk. Also the intrinsic risks can be induced by the vagueness on a level geometric attribute (*Risk-Geometry*) or by the aggregation formula (*Risk-Aggregation*). From there, we have proposed a tolerance scale, composed of four (4) levels (*0-totally unacceptable risk, 1-preferably unacceptable risk, 2-somewhat acceptable risk, 3-totally acceptable risk*) to help end-users express their sensibility to any identified risk. Then we have associated to each tolerance level a risk management strategy and possible corresponding actions producers can choose from to reduce the risks (e.g. *delete a level present an unacceptable risk, communicate a risk via visualization policies, and modify an aggregator*). Finally, we have worked out a new general risk-aware SOLAP datacubes design process by integrating a risk management method to the classic design process (requirement specification and conceptual design phases to be more precise). The results of this phase are compiled in a paper that can be found also at the chapter 3 of this document.

Phase 3: Proposal of a rapid prototyping risk-aware design method called the RADSOLAP method

There are different multidimensional design methods:(Böhnlein, Plaha et al. 2002, Lujan-Mora, Trujillo et al. 2006, Prat, Akoka et al. 2006);(Jensen, Holmgren et al. 2004, Guimond 2005, Mazón, Trujillo et al. 2007, Malinowski and Zimányi 2008, Romero and Abelló 2008, Song, Piattini et al. 2008, Romero and Abelló 2010). These methods are all based on user-driven, sources-driven or hybrid approaches. In our work, we consider a new aspect related to risks of misinterpretation of the datacubes, therefore we are interested in a method that is based on a hybrid approach but is also risk-aware. Such a method is yet to be proposed in the literature. Thus in this phase, we work towards elaborating such a risk-aware multidimensional design method using the results of the previous phase 2. We quickly realized that risk-awareness and risk management calls for many multidimensional model transformations during the design process and additional variables that are risk communication policies. We also realized that it is important for SOLAP datacube producers to quickly obtain SOLAP datacube prototypes in order to validate the design with end-users by real testing with sample data. A rapid prototyping method is thus required. The RADSOLAP method proposed to answer all three requirements (hybrid, risk-aware and prototyping) is the extension of a ProtOLAP (Bimonte, Nazih et al. 2013) method elaborated by our research group at Irstea with our collaboration (Cf. middle corridor of Figure 1-2). The ProtOLAP method is a rapid OLAP prototyping with on-demand data supply method. It is based on the UML formalism for the

multidimensional and aggregation modeling; also it is semi-automatic since it allows generating logical schemas for the OLAP server and the DBMS from the multidimensional model. Furthermore it is supported by a technical CASE tool developed by the research team. Our extension activities have been the followings:

- *Extending the ProtOLAP design process* to implement the risk-aware method developed in our second phase.

- *Extending the UML profile* designed by Boulil, Bimonte et al. (2012) to model SOLAP datacubes in order to take into account our new geographic data typology, the risk and the risk communication variables and the tolerance levels.

- *Formalizing multidimensional model transformation functions* in order to insure that the model is still valid throughout the design. We have formalized for example the functions *DeleteLowestLevel*, *DeleteDimension* and *ModifyAggregator*. The formalization language is based on the UML profile elements.

- *Defining the transformation process* to help datacube producers insure the correctness and the coherence of all the transformations. This process consists of three steps which are the splitting step, the transformation step (where the functions are applied) and the fusion step.

The results of this phase, which allow us achieving our sub-objective 2, are also compiled in a paper that can be found at the Chapter 4 of this document.

Phase 4: Validation of the thesis proposals

This phase is where we implement our proposals and make use of them on a real case study in order to verify our initial research hypothesis. This phase meets sub-objectives 3 and 4. It includes the following stages:

- *Definition and implementation of a CASE Tool to support the RADSOLAP method*:
  As explained in the phase 3, the ProtOLAP research team at Irstea has already worked on a CASE Tool to support ProtOLAP. The tool architecture is composed of four tiers namely: the Requirement Tier (where the conceptual design is done using the designed UML profile), the Schema Tier (where the logical schemas for the OLAP server and the DBMS are generated), the Feeding Tier (where end-users can feed the OLAP datacube with sample data) and the Visualization Tier (where users can explore the datacube via an OLAP client). For the extended tool, called SOLAP RADTool, we have added another tier, the Transformation Tier, where SOLAP datacubes models transformations are done. The Transformation Tier was implemented by two undergrad students under our supervision as part of a 120h student project.

- *Testing on a real case study:*

  Since the new risk-aware design approach and the datacube usability strongly rely on the end-users, the best way to analyze and understand the efficiency of our method is to use a case study. For that reason, we have chosen a case study related to the sewage sludge spreading activities. Irstea Clermont-Ferrand has been in charge of a national Web GIS-based system (called SIGEMO/SILLAGE) dedicated to the planning of agricultural spreading of sewage sludge produced by French farmers and wastewater plants. It contains vector and numerical data on spreading activities (farms, agricultural plots and spread zones etc.) from 2006 to 2013. The areas unsuitable for spreading activities are drawn on maps by users (e.g., farmers) via the Web-based system as, for example, buffers around water bodies or buildings. Spread zones are therefore the agricultural plots excluding the unsuitable areas. Such zones have vague shapes for two reasons: the excluding water bodies have vague shapes and the spreading equipment and methods cause an offset (uncertainty) between the spread zones limits drawn and the real ones. We have defined a multidimensional model for a datacube exploiting the spread zones using our extend UML profile. Then we have applied the RADSOLAP method with the ultimate goal of verifying that the method provides usable and reliable SOLAP datacubes that can be easily implemented in existing classical tools. That means: (1) risks of misinterpretation (Risk-Aggregation and Risk-Geometry) were determined knowing the vagueness present in the initial SOLAP datacube multidimensional model, (2) all possible combinations of tolerance levels were associated to the risks identified alongside with risks management actions; (3) and finally the initial multidimensional model have been transformed according to the tolerance levels using the RADTool (Transformation Tier).

- *Verifying designed SOLAP datacubes usability, reliability and ease of implementation in classical SOLAP architectures:*

  In this step, we compare the SOLAP datacubes produced in the previous step with the SOLAP datacube resulting from a classical design method. First we evaluate the usability, reliability and ease of implementation of each produced SOLAP datacube and then we compare the evaluation results. Because our work focus is put on the SOLAP datacube multidimensional structures, we concentrate the usability testing on the multidimensional structure testing. There are few proposals in the literature regarding datacubes multidimensional structure usability tests; the testing methods are in majority based on quantitative metrics well defined and validated through experimentation. Since it is not our objective to propose testing method and/or new metrics, we have based our usability testing on the existing literature (Berenguer, Romero et al. 2005, Serrano, Trujillo et al. 2007, Golfarelli and Rizzi 2011) by choosing meaningful existing metrics (such as number of

measures, number of multiple hierarchies etc.) from that literature to evaluate the usability of all the produced SOLAP datacubes multidimensional structures.

Concerning the reliability testing, we need to evaluate if the spatial vagueness is considered or not for all the resulting SOLAP datacubes. In our method, the spatial vagueness was recognized and considered for all the produced SOLAP datacubes since at some point of the design the vague objects were identified and translated into corresponding risks of misinterpretation. Therefore we have essentially focus the reliability testing on highlighting objectively how the spatial vagueness is actually considered through our method and how nothing is done in that sense in the classical methods.

Regarding the ease of implementation criteria, actually the resulting SOLAP datacubes are technically the same. They all use crisp representation of the spatial data and no new data type or SOLAP operator has been introduced. Moreover, we have integrated the implementation of our method in a classical CASE tool ecosystem which allow the automatic generation of classical schemata for the SOLAP servers and SDBMS. So the ease of implementation has simply been evaluated and compared through a discussion.

The results of this fourth phase, which allow us achieving our sub-objectives 3 and 4, are detailed in the chapter 5 of this document.

The following activity diagram describes our methodology (Cf. Figure 1-2). Note that there are three different corridors in this diagram: the first one presents the activities conducted alone; the second presents the results of our collaboration with research team members; and the third one presents the results of our collaboration with undergrad students through students' projects.

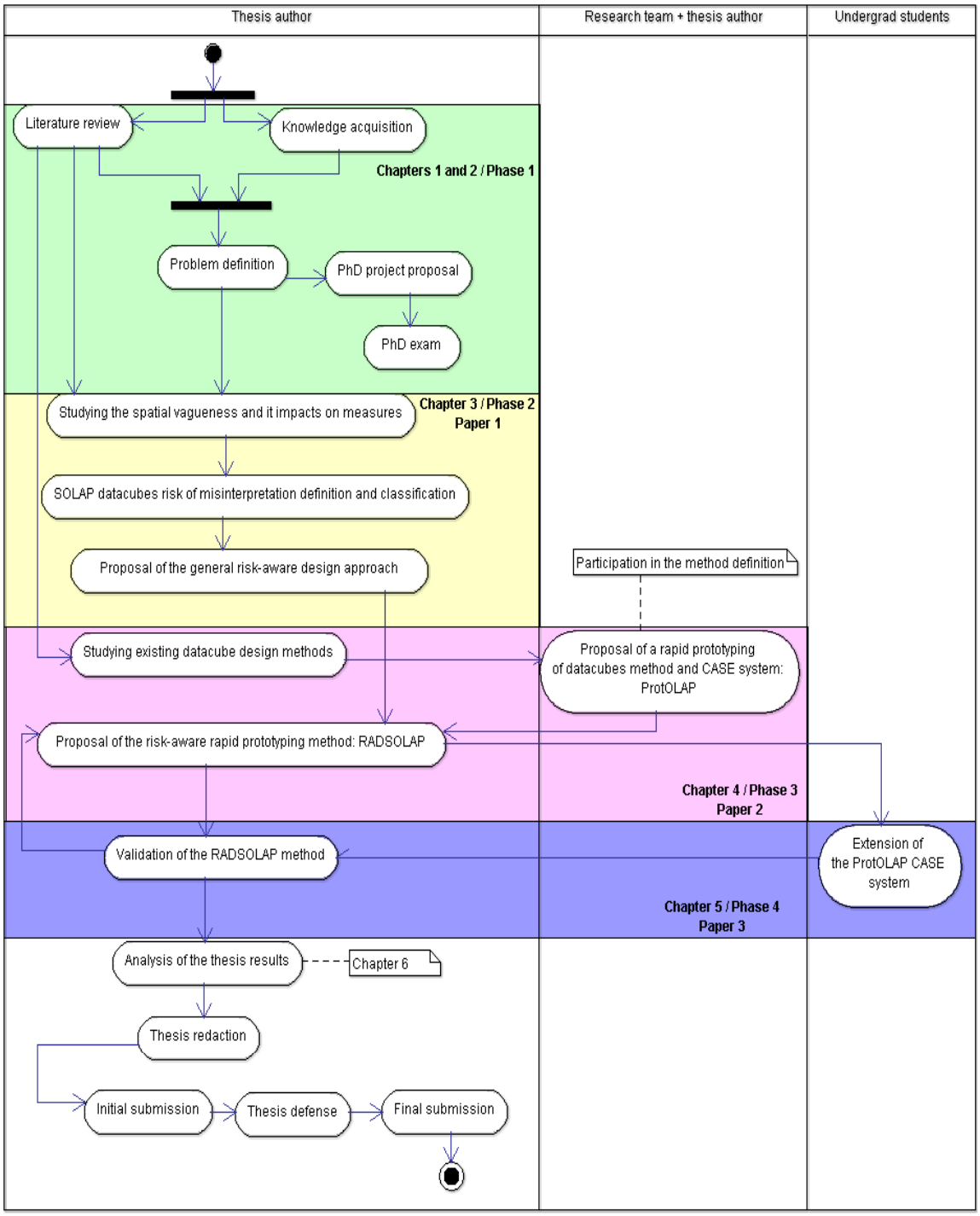


Figure 1-2: UML activity diagram of the research methodology and thesis main steps

1.6 Thesis positioning in the Irstea and Laval University research on spatial data quality landscape

This thesis has been realized in the context of an unofficial global project dealing with quality issue in spatial databases and spatial datacubes and web geoportals. This global project has been going on for few years now and different M.Sc. and Ph.D. students from our research groups from Irstea in France and Centre of Research in Geomatics (CRG) in Canada have participated in it. The research objectives were thought to be different but complementary.

Let us start with the work carried out at the Centre of Research in Geomatics by seven of our colleagues: Ph.D. students Amaneh Jadidi Mardkheh, Joel Grira, Tarek Sboui, and Mehrdad Salehi; M.Sc. students Marie-André Lévesque and Tania Roy.

Amaneh Jadidi Mardkheh work has been presented in section 1.2of the present document. In summary, she proposed fundamental tools required to represent and exploit regions with fuzzy semantic, temporality and vague shapes in spatial datacubes. She based her contributions on the fuzzy logic and applied them on coastal erosion risk assessment with the multidimensional paradigm. She defended her thesis in March 2014. Although we both are dealing with spatial vague objects exploitation in datacubes, our goals are different but complementary. In fact, our work aim at offering a solution that manage the risks of misinterpretation related to the exploitation of vague objects in classic SOLAP datacubes while her work aims at offering tools that help elaborate and analyze fuzzy SOLAP datacubes. Her solution can be the ultimate one for end-users that do not tolerate any identified risk in our approach.

Marie-André Lévesque focused on proposing a formal approach for a better risk of misuse identification and management in SOLAP datacubes. More specifically, she proposed a SOLAP datacube risk of misuse definition in accordance with the ISO (2000) definition of risk, as well as a risk of misuse classification and forms to help identify and describe the risks of misuse. She also adapted the risk management method and integrated it to the SOLAP datacube design process. Finally, she proposed a risk reduction approach which consists of communicating the risks via popping context-sensitive warnings in some multidimensional queries (Lévesque 2008). Her contributions offer the fundamentals for the SOLAP risk of misuse definition, description and management in a preventive approach. Unlike her solution, our solution not only aims at addressing the risk related to spatial vagueness specifically, but also at making the multidimensional elements definition aware of the end-users tolerance to the identified risks. In addition, we aim at offering a semi-automatic solution. Note that her work was over in 2008.

Joel Grira introduced risk management in the database design process regardless the decisional platform (GIS, SOLAP systems etc.). He proposed a collaborative approach based on crowdsourcing technology to identify and analyze potential risks of data misuses (Grira, Bédard et al. 2013). The approach relies on end-users' feedbacks about the ways the elements (object class, property, function, association, domains)

of a conceptual database design are defined. He also proposes collaborative tools such as wiki, questionnaire and forum to find the identified risks for the given elements definitions and to improve these definitions if decided so. For a given identified risk, the database design team can afterwards select an appropriate risk management strategy, just as advocated in our own work. In this regard, his work is also complementary to ours. For the record, he defended his thesis in May 2014.

Tania Roy proposed an a posteriori approach to help end-users (non-experts of spatial data) of an already existing system, namely the geoportals, identify and assess the potential risks of misuse they incur. The approach is based on a series of structured questions users have to answer in order to identify the risks. Different possible risk management actions are then determined according to the risks identified. She also offered recommendations to data producers in regards with the data fitness for use and risks of misuse communication to end-users (Roy 2013).While her solution is meant for already existing systems, the geoportals, ours aims at taking the risks into account during the systems (SOLAP datacubes) production (a priori solution). For the record, this master thesis was defended at the end of 2013.

Tarek Sboui focused his research on the interoperability between two SOLAP datacubes. He proposed a conceptual framework to deal with the semantic heterogeneity issues between datacubes as well as a systematic approach to manage the risks related to semantic interoperability data misinterpretation. His solution consisted of a set of indicators which allow identifying and assessing the risks in addition with a framework that help the relevant stakeholders make decisions relative to the risks (Sboui 2010). His work was over in 2010. The fundamental quality issue is different since we are not working on semantic interoperability but spatial vagueness. However, the idea of a set of indicators to identify and assess the risks is interesting and can be used in complementary of our proposition.

Finally, Mehrdad Salehi on his part focused his work on defining and classifying different types of integrity constraints in spatial datacubes. First, he proposed a required formal model for spatial datacubes where he defines the different elements of the spatial datacube multidimensional structure (spatial measure, spatial dimension, spatial fact etc.). Then, based on that model, he identified the different integrity constraints, before classifying them in categories and sub-categories (e.g. fact integrity constraints, traditional integrity constraints, summarizability integrity constraints). He also developed a formal integrity constraints specification language (ICSL) based on a controlled natural language and a natural hybrid language with pictograms (Salehi 2009). Our works are completely different but they come together on the quality of produced spatial datacubes aspect. Also, integrity constraints can be exploited in our approach to prevent some of the risks of misinterpretation.

Now, regarding the works carried out in Irstea, Clermont-Ferrand centre to be precise, two Ph.D. students have tackled two different aspects of the quality issue: Kamal Boulil for the integrity constraints in spatial datacubes and Lotfi Bejaoui for the spatial vagueness.

Kamal Boulil work covered the definition and implementation of semantic integrity constraints in SOLAP systems based on the Spatial OCL[2] standard and the UML language. His work is in the continuity of Mehrdad Salehi presented right above. One of his contributions is an UML Profile that allow datacube designers defining not only the multidimensional structure of SOLAP datacubes (core model) but also the aggregations model expressing end-users' needs in analysis (Boulil 2012) as well as appropriate integrity constraints on both groups of elements. Our work addresses different aspects of the quality issue in SOLAP systems but his UML Profile has been a good tool to be exploited in our approach in order to allow datacube producers to define valid visual datacube model plus metadata of risks of misinterpretation and related tolerance values. More details are given on this particular point in the Chapter 4: of this thesis. This thesis was defended at the end of 2012.

Finally, Lotfi Bejaoui proposed an exact model (the Qualitative Min-Max model) with topological relationships qualification to represent and exploit objects with vague shapes in spatial databases and spatial datacubes (Bejaoui 2009). In general, his work is of equal interest to us as Jadidi's fuzzy logic approach, meaning that is it useful to offer a solution to end-users that do not tolerate any of the risks identified. However, we base our spatial vague objects conception on the Qualitative Min-Max model in our work. His thesis jointly supervised by the CRG, and Irstea Clermont-Ferrand was defended in 2009.More details on how the model influences our work are given mainly in Chapter 3:.

1.7 Structure of the thesis

This thesis has six chapters. **Chapter 1** introduces the problem, the research question and the methodology adopted to answer this question. **Chapter 2** presents main general concepts related to the research areas targeted in this thesis: spatial data uncertainty and quality, SOLAP systems, spatial vagueness management in SOLAP systems, risks of misuse management and multidimensional design method. Chapter 3, 4 and 5 detail not only the literature review briefly presented in this first chapter, but also our main contributions. **Chapter 3** is where the general risk-aware design approach is described. In a first part, the risk of misinterpretation is defined and classified and a tolerance scale is defined as well as a set of possible risk management actions corresponding to each tolerance level. In a second part, the geometric data typology adopted for the symbiotic trade-off approach as well as the analyses of the spatial vagueness impacts on the multidimensional are also presented here. The first part, which is the main content of this chapter, was submitted and accepted as a paper in a journal with blind peer-reviewed. In **Chapter 4**, the focus is on the rapid risk-aware prototyping of SOLAP datacube method implementing the results detailed in Chapter 3. In particular, the UML profile extension, the datacube model transformations functions and the transformation process are explained. This chapter was also submitted and accepted in a journal. In C**hapter 5**, we are interested in the validation of the research results. The ProtOLAP system

[2] Object Constraint Language

extension was briefly presented and we provide details on the validation method in addition with the results (usability testing, discussion on reliability and ease of implementation). This chapter has also been submitted and accepted in a peer-review conference and extended for a special issue of a peer-review journal. The extended version is the one presented in this thesis. Finally in **Chapter 6**, we review our contributions and provide a discussion for each one of them. This chapter also present our research perspectives.

Chapter 2: Concepts review

2.1 Introduction

This chapter focuses on the concepts related to the areas covered in this thesis, which are spatial data uncertainty, Spatial OLAP systems and to a lesser extent the management of datacubes risks of misuse. At first, spatial data uncertainty, in particular the spatial vagueness, and spatial data quality are defined and explained (section 2.2). Then we present Spatial OLAP systems fundamentals (section 2.3) and finally notions related to the management of datacube risks of misuse are provided (section 2.5).

2.2 Spatial data uncertainty and spatial data quality

2.2.1 Uncertainty in spatial data

Spatial data is an object with a spatial reference and descriptive attributes (Salehi, Bédard et al. 2010). The spatial reference is usually a shape and/or position described textually (postal code, address, etc.) or with spatial coordinates (X, Y).The spatial reference and attributes acquisition is done by spatial data experts by means of specific communication and modeling processes, techniques and instruments.

Spatial data is often subject to uncertainty due to the whole acquisition and dissemination process. Indeed, the spatial data modeling process, the acquisition and processing technologies and/or the people intervening at different levels of the spatial data life cycle (modeling, acquisition, processing, communication/dissemination and usage) tend to create different types of uncertainty not only on measurements of attributes, space and time but also on the identification and classification of objects. The uncertainty can be for instance an inaccuracy, a fuzziness, a spatial vagueness, etc.

Fisher (1999) and Bédard (1986) have each modeled and classified the spatial data uncertainty from two different points of view. The combination of both their work allows us to well describe and understand the spatial vagueness sources and types in the next paragraphs.

For Bédard (1986), the uncertainty introduced in the spatial data comes from two main sources: (1) the inherent limitations of the modeling process during the models production or communication and (2) the limitations relative to reality observers and model users.

(1) *Regarding the inherent limitations of the modeling process*: According to the author, the most important limitation is probably the fact that the models are simplified and approximate estimations of the reality in respect of the usage goal and context. This limitation translates into

vagueness in the identification and labeling of the spatial objects and also into limitations in those objects properties measurements. Indeed, vagueness in the identification and the classification occurs for example when people classify spatial objects with broad semantic or physical boundaries into discrete groups of entities. It is not possible to determine a sharp and univocal separation between two objects with broad semantic or physical boundaries (example of forests areas: when does it become such an area? when are two such areas distinct? when does a forest area becomes a field with individual trees?) thus unfortunately a discrete classification induces an uncertainty on the existence of the object and on the classification in the right group even if the object attributes measurements are done with the highest accuracy (to be or not to be a forest area? to be a forest area or a field?). The answers are goal-dependent and context-dependent. Higher certainty requires more explicit ontologies and more precise measurement devices and methods.

(2) *Regarding the limitations relative to model-makers and users:* it concerns the people involved in the spatial data creation and communication, namely the real world observers, the intermediate people that transfer the data and the final users. The first main idea here is that model-makers introduce subjectivity into the spatial data no matter how precise, correct and well-quantified we think they are. In fact, the reality models we produce, both cognitive and physical, are subject to our reference framework fed by our own history, education, needs, experiences, cultural, professional and familial environment etc. Therefore, for a same reality, the produced cognitive model is different for each person or even for the same person at different periods, for different contexts and for different needs. So are physical models, to a lesser degree. The second main idea is that instead of analyzing all the different possibilities in order to choose the best real world representation, model-makers are satisfied with the first good alternative that respond to the needs. This is called the concept of «satisficing». Consequently, the spatial data produced is not always reliable for every usage.

The uncertainty relative to both limitations is inevitable and it is clear that we cannot have a perfect representation of the real world in our mental models and in their physical representations.

In view of the above, Bédard (1986) proposes a categorization of the different kinds of spatial data uncertainty into four orders:

- The first order (*conceptual*) refers to the uncertainty on the identification of an observed reality (What do we call building? Is this a building? If yes, is it Commercial or type Residential?);

- The second order (*descriptive*) refers to the uncertainty on the attribute values: measurement error or imprecision in the quantitative values, or fuzziness on the qualitative values (e.g. rich soil);

- The third order (*location*) refers to the uncertainty on the localization of the observed reality in space and time (e.g. shape vagueness, fuzzy boundaries, imprecise positioning because of a lack of information);

- The fourth order (*meta-uncertainty*) refers to the degree an uncertainty of one of the previous three levels is unknown.

For Fisher (1999) spatial objects (observed phenomena) can be classified in two categories: spatial objects that allow an unequivocal separation into discrete classes (well-defined objects) and spatial objects for which it is not possible to do that (ill or poorly defined objects). Well defined spatial objects are essentially geographies defined by humans (and associated with attributes) such as census areas, land ownership and administrative divisions to name a few (N.B. such objects are better defined, but not without Bédard's four orders of uncertainty). Such objects limits are often marked on the ground (naturally or by humans), but not always (ex. census areas). Examples of ill-defined objects are trees stands, vegetation and soil occupations. In the three cases, the existence of intergrades makes it difficult to determine the entities to be mapped and the spaces they occupy. Majority of natural and built environments objects are ill-defined.

The uncertainty differs whether the spatial object is well or ill defined. Indeed as described in his model of uncertainty (see Figure 2-1), when the spatial object is ill-defined, the uncertainty is due to the vagueness (When a forest is a forest?) or the ambiguity (discord or non-specificity) in its definition/conception. Whereas the definition vagueness is inherent to the spatial object to be modeled, the ambiguity occurs "*when there is doubt as to how the phenomenon should be classified because of differing perceptions of it*" (Fisher 1999). Just as the perceptions one have of the phenomenon, the ambiguity is context-dependent and model-dependent. For example, for a given modeler, an object can be perceived and classified as a building when for another modeler, the object can be ignored or classified as something else because of a non-specificity in the building definition (the first one maps for example all the buildings while the other one maps only the buildings that have a roof perimeter greater than $20m^2$).

When the object is well-defined, the uncertainty is caused by errors (e.g. measurements errors, errors in spatial generalization and labeling errors).

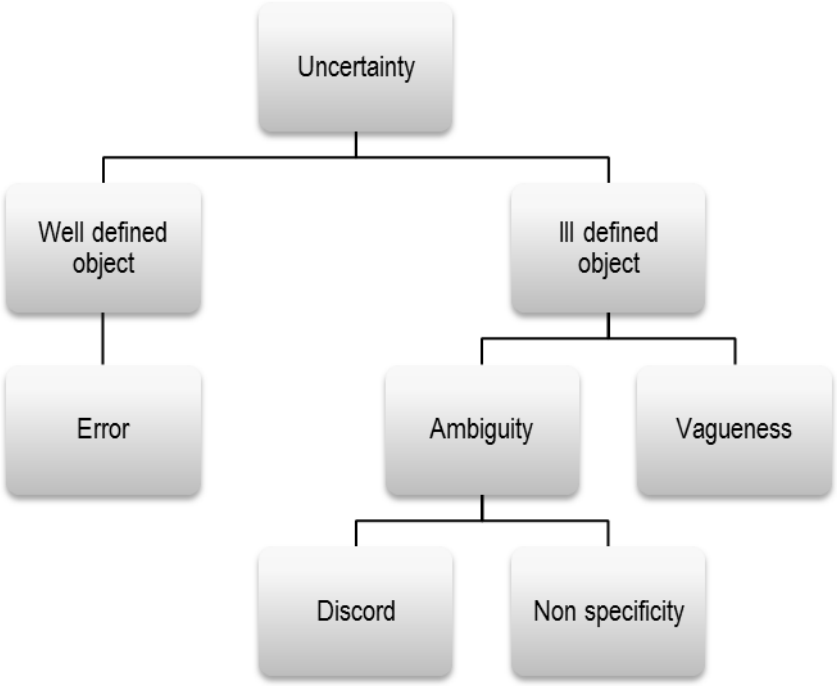


Figure 2-1 : Conceptual model of uncertainty in spatial data (Fisher 1999)

To properly describe exhaustively the types of uncertainty by benefiting from these two models, we have merged them keeping the entry points of Fisher (1999) (Well and ill-defined objects) and then classifying the uncertainty types arising under Bédard (1986) uncertainty orders (see Figure 2-2).

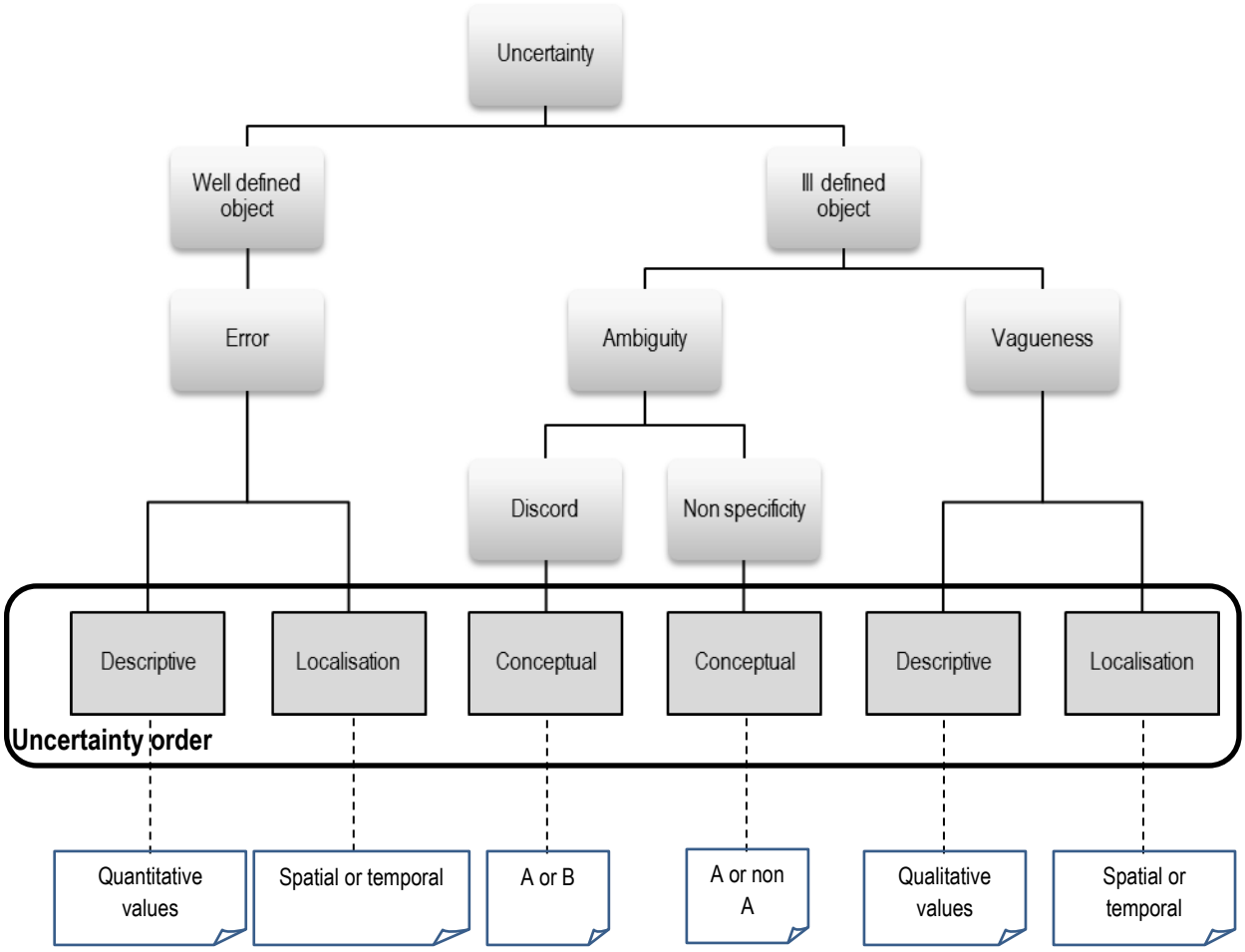


Figure 2-2: Model of uncertainty in spatial data: Fisher (1999) and Bédard (1986) uncertainty models.

From this model, it is obvious that spatial objects vague conception issues lead to descriptive, spatial and/or temporal uncertainty on the spatial data while ambiguity issues lead eventually to conceptual uncertainty. Conceptual uncertainty can in turn introduce spatial uncertainty on spatial data.

Also we can conclude from both Fisher (1999) and Bédard (1986) contributions that ill-definition is not only a matter of positioning (unknown location), but also a matter of vague conception of the entity (e.g. what do we call ocean?), fuzziness in its identification (e.g. when a tree is a tree?) or fuzziness in its descriptive qualitative attributes (e.g. poor, rich or very rich soil).

2.2.2 Definition of spatial vagueness

Spatial vagueness is an uncertainty on spatial data that can be categorized into *shape vagueness* and *location vagueness*:

- *Shape vagueness* designates an uncertainty on the shape of an object that represents an ill-defined real object (e.g. forests, lakes with islets, flood zones) (Bejaoui, Pinet et al. 2009). It occurs when for the sake of simplification, the objects, which have broad boundaries or are vaguely defined, are represented by discrete polygons, lines and points.

- *Location vagueness* is either an uncertainty on the position of the spatial object (resulting from a lack of knowledge about the position of an object with an existing sharp boundary) or an uncertainty on the measurement (resulting from the inability to measure such an object precisely) (Schneider 1999, Hazarika and Cohn 2001). E.g. a river portion in an unreachable place.

2.2.3 Quality concepts

Spatial data quality concept is another aspect of the uncertainty issues in spatial data. In contrary to the uncertainty concept, quality concept is relative to requirements and usually associated to standards. To determine the quality of a set of spatial data, it is possible to use the metadata or to have a comparison referential which is the "universe of discourse". Usually, the quality evaluation is done using the universe of discourse because metadata are not always or fully provided. The universe of discourse (Devillers and Jeansoulin 2006) is the ideal set of data that correspond to the specifications or the user's requirements. In practice, the universe of discourse consist of a data set which is known for having a better quality (e.g. ortho-photo, another data provider) because the ideal, perfect spatial data cannot be acquired. The standards offer quality criteria that one can exploit to evaluate the quality of a data set using the set of reference.

Because it is a relative concept, there is not a unique definition of the quality. For some, a quality product is a product free of errors and that fits the specifications; for others, it is a product that fits users' expectations. Those two points of view are translated into the two concepts of internal quality and external quality.

Internal quality (ISO/TC 211 2002): it corresponds to the level of similarity existing between produced data and perfect data (universe of discourse). The criteria to evaluate the internal quality are:

- The *completeness*: is the data set exhaustive (all the entities are present)? Do we have missing attributes or relations? Do we have entities, attributes or relations that should not be present in the dataset according to the specifications? This criteria indicators are the commission (non-expected presence of entities, attributes or relations) and omission (absence of entities, attributes or relations) degree. It can be evaluated for instance by calculating the percentage of forests or the percentage of null road names in a dataset.

- The *position accuracy* (also known as geometric precision or spatial accuracy): It refers to the objects' position precision. The accuracy can be expressed in absolute or relatively to the set

of coordinates known to be true (reference dataset). It can be indicated by the mean square error on the positions. E.g. the agricultural plots are positioned with a precision of 5m.

- The *attribute accuracy* (also designated as non-spatial attribute precision (Servigne, Lesage et al. 2005)): it refers to the quantitative attributes accuracy and the correctness of the non-quantitative attributes. E.g. the agricultural plots areas are calculated with a precision of $5m^2$.

- *The thematic accuracy*: it refers to the correctness of the entities and relations classification. The question to answer is for instance "Are the buildings correctly classified?" One way to answer such question is to calculate the confusion matrix; another way is to compare the classification to the classification done in the reference dataset.

- The *temporal accuracy* (or temporal precision): it refers to the precision of a time measurement, the temporal consistency and validity, and the accuracy of entities temporal attributes and relationships. Example of verification: Are the roads up to date?

- The *logical consistency*: it refers to the degree of adherence to the logical rules of data structure, attributes or relationships. It includes the conceptual consistency (Do the objects in the database and the relationships between the objects comply with the conceptual schema and rules?); the values domain consistency (Do the attributes adhere to the appropriate values?); the format consistency (Are the data stored in the appropriate format?); and the topological consistency (Are the topological relationships represented and true to the reality?). E.g. the trees must be represented only if the foliage diameter is greater than 5m.

External quality or ***"fitness for use"***(Bédard 1995): it corresponds to the similarity level existing between the produced data and the users' needs/expectations. To evaluate the external quality of a spatial data set, it is possible to use criteria such as *accessibility* (data set cost, format, rights etc.), *genealogy* (acquisition methods, producers etc.), *precision* (semantic, temporal, spatial precision relatively to user's needs), *definition* (evaluation of the exact nature of the data and the object it represents relatively to the needs on the semantic, spatial and temporal aspects) or *legitimacy* (compliance to standards etc.).

A dataset can present uncertainty issues but still of a good quality, internal or external, as long as it satisfies the specifications/users' needs. For instance, a forests dataset which present shape vagueness issues might still suit most of users for whom it is enough to have open data, up to date with a spatial precision of 5m and for which all the name attributes are filled.

2.3 Spatial OLAP Systems

Even though GIS are powerful for spatial analysis and map display, they are built for transactional purpose and are not suitable for a rapid and interactive spatio-temporal multidimensional analysis. Therefore, they

have been coupled with the On-Line Analytical Processing technologies in the mid-90 (Bédard, Larrivée et al. 1997, Stefanovic 1997, Caron 1998, Han, Stefanovic et al. 1998) to make profit of both technologies. GIS bring powerful spatial analysis tools and cartographic visualization while the OLAP systems bring tools for large volume of data interactive exploration and analysis. The coupling results in Spatial OLAP (SOLAP), defined as a «visual platform specially designed to support easy and rapid spatio-temporal analysis and data mining in a multidimensional approach based on aggregation levels, and to allow cartographic, graphical and tabular displays» (Bédard, Merrett et al. 2001). End-users of SOLAP systems can visually detect unknown patterns and spatial phenomena and verify and/or formulate hypotheses via a spatio-temporal easy and interactive analysis of their data.

In section 2.3.1 and 2.3.2 we describe the spatio-multidimensional model which is, in most cases, the heart of SOLAP systems; then we present the architecture and tools supporting SOLAP platforms in section 2.3.3; finally, SOLAP datacubes production process and challenges are presented in section 2.3.4.

2.3.1 The spatio-multidimensional model

In this thesis, we are interested in spatial data with vector representation (and not matrix one) therefore the concepts described here are based on a vector representation of the geometric attributes.

The multidimensional model is based on new key concepts such as the "*dimension*", "*measure*", "*fact*" etc. (Bédard, Rivest et al. 2006).

Dimensions are the analysis themes (e.g. Product, Time, Location, Clients). They contain *"members"* organized in *"hierarchies"* according to their granularity level (e.g. hierarchy level Day containing members such as 01-02-2014; Day is grouped in Month, grouped in turn in Season, grouped in turn in Year). The members of a low level (e.g. Month) are usually aggregated to constitute the members of a superior level (e.g. Season) in the hierarchy.

The ***measures*** *are* the analyzed values (e.g. number of products sold, sales turnover) associated to each combination of members belonging to the different dimensions. They reflect the state of a situation in relation to the dimensions of analysis. They are dependent variables that constitute the facts attributes alongside with the members of dimensions. Example: number of agricultural plots Nb_agricultural_plots.

Facts are the analysis subjects represented by the combination of measures and members of dimensions. It is for example "There are "5200" agricultural plots (Nb_agricultural_plots) cultivated in "2004" (Time) for the "Auvergne" (Location) region". The facts are stored in a fact table that is queried afterwards.

The dimensions instances plus the facts form the stored datacube.

Datacubes are the physical implementation of the multidimensional model. They are actually hypercubes with 2 or more dimensions.

The extension of the multidimensional model with the spatial component results in the definition of the spatio-multidimensional elements (Bimonte, Wehrle et al. 2006, Salehi, Bédard et al. 2010):

A **spatial dimension** is a dimension presenting at least one hierarchical level containing a spatial member (the level is thus a spatial level). A *"spatial member"* is an object with a spatial reference; it can be geometrical (having a geometry) or descriptive (a country name for example). In any case, there should be a semantic relationship between members of two spatial levels of a same spatial hierarchy and in most cases, not always, there is an intersection or inclusion relationship between the members.

There are three types of spatial dimensions: the *"non-geometric spatial dimension"* that only holds descriptive spatial members; the *"geometric spatial dimension"* where each member possess a geometry (most of the time polygonal geometries) which allows the cartographic visualization, the spatial drill-down or other spatial operations on the members (Proulx and Bédard 2004); the *"mixed spatial dimension"* which contains a combination of descriptive and geometrical members.

A **spatial measure** is a measure with a spatial reference. It can be a numerical value ("*numeric spatial measure*") resulting from spatial data processing (e.g. calculated area of a plot Agricultural_plots_surface) (Rivest, Bédard et al. 2001, Malinowski and Zimányi 2004), or a set of coordinates or pointers on geographic primitives (*"geometric spatial measure"*) resulting from spatial operation such as an intersection or a union between two geometric spatial dimensions (e.g. position of buildings subjects to flood risk resulting from the intersection between the flood risk areas and the buildings layer)(Stefanovic, Han et al. 2000, Rivest, Bédard et al. 2001, Malinowski and Zimányi 2004, Sampaio, Sousa et al. 2006). It can also be a *"mixed spatial measure"* with a combination of a numeric value and geometry (e.g. agricultural plots number associated with the plots localizations) (Bédard, Merrett et al. 2001).A spatial measure, when observed at coarser hierarchical levels, is aggregated using classical SQL aggregation functions such as SUM, MIN, MAX, etc. or specific functions that can be operated on spatial data (union, intersection, union + area, intersection + length etc.) either from the lower level or from the finest-grained level.

A **spatial fact** (*"non-geometric"* or *"geometric"*) describes *an event of interest for a decision-making process that happened in the space* according to Salehi, Bédard et al. (2010). A spatial fact is geometric when it contains at least one geometric spatial measure and one geometric spatial member otherwise it is a non-geometric spatial fact. E.g. "The total number of agricultural plots cultivated in "2010" (Time) in "Auvergne" (Location) is "402" (Nb_agricultural_plots)" associated to a multipolygon representing the union of the plots extents and positions which can be visualized on a map.

A **spatial datacube** is known as a datacube where *"certain facts or members of dimensions have a spatial reference and can be represented on a map"* (Bédard, Merrett et al. 2001).In other words, it is a datacube that has at least one geometric spatial fact (meaning a geometric measure or mixed spatial dimension instances. Technically, a datacube that has only spatial but non-geometric instances of spatial facts and dimensions is still a spatial datacube. However the common understanding of spatial datacube, especially in the Computer Science community, is that it is a datacube which supplies a cartographic representation end-users can exploit for data visualization and exploration.

In this thesis, we designate spatial datacube, as defined above, by SOLAP datacube in order to avoid the interrogations on the term "spatial". Also, when mentioning spatial dimension we will be talking about geometric spatial dimension composed with geometric members.

Using this SOLAP datacube, end-users do not need to know the database structure or any query language to quickly (few seconds) answer questions like "What was the total surface of agricultural plots cultivated in Auvergne in 2000?" In fact, thanks to SOLAP systems components, the data are presented to the end-users in pivot tables, diagrams and on maps (see Figure 2-6); therefore by simple clicks on tables and maps, they navigate through them.

2.3.2 Spatial hierarchies

A spatial level can be geometric or non-geometric (cf. spatial dimension definition in section 2.3.1) and spatial levels can be organized into a strict or a non-strict spatial hierarchy (Zimányi and Malinowski 2008).

A **strict spatial hierarchy** is a spatial hierarchy where only one-to-many relationships can be found between the levels. That means each child member is related to at most one parent member and a parent member may be related to many child members in the hierarchy. For example city Clermont-Ferrand is located in the region Auvergne and the region Auvergne strictly contains Clermont-Ferrand, in addition with Aubière, Aurillac, and Moulins etc.

A **spatial hierarchy** is said to be non-strict when it has at least one many-to-many relationship between the levels (Zimányi and Malinowski 2008). For example in a hierarchy Agricultural plots < Watersheds < Country, a watershed is related to many different agricultural plots and an agricultural plot may belong (overlap) to two different watersheds.

In either case, it is important to define the appropriate topological relationship between consecutive spatial levels to ensure correct aggregation of measures in higher levels. In general, according to the Calculus-Based Method (CBM) implemented in PostGIS (Clementini, Felice et al. 1993), there are five basic possible topological relationships between two single geometries: ***touch*** *(the interior intersection is empty but the boundaries intersection is not)*, ***disjoint*** (the intersection is empty), ***in*** *(the intersection result in one of the two geometries)*, ***cross*** *(between line/line and line/polygon only),* and ***overlap*** *(applicable to two*

polygons or two lines; the intersection is not empty and is not equal to one of the two geometries). The relationships are exclusive, meaning that it is not possible to have two relationships for two single geometries at the same time. The Figure 2-3 illustrates the basic topological relationships between two polygons.

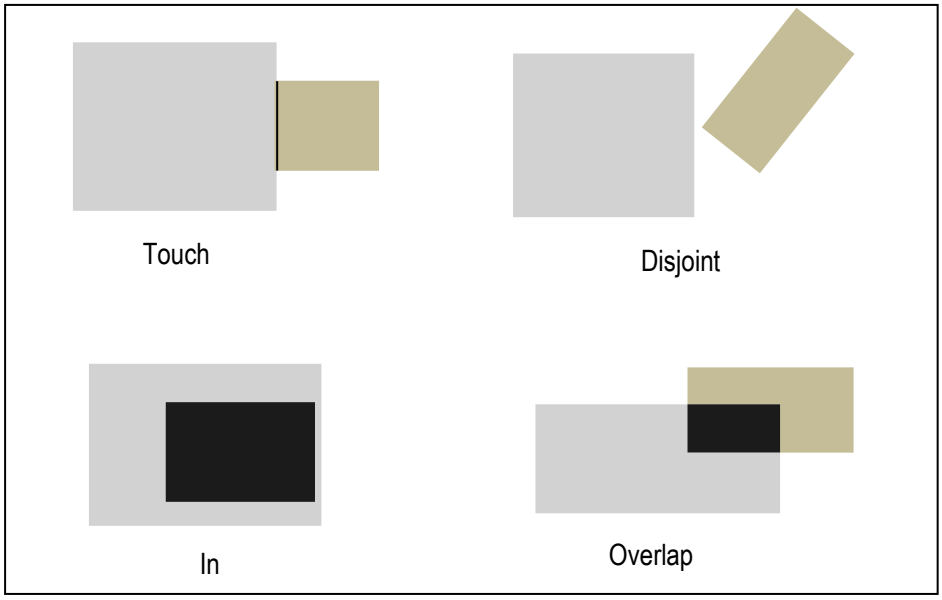


Figure 2-3: Illustration of the topological relationships between two single polygons

Depending on the topological relationship, the aggregation can be done in a classic way or it can need a specific procedure. When the geometry representing the spatial union of the child members is in the parent member geometry, the aggregation is classic. We consider for example the cities child-members of the administrative department Puy-De-Dôme and the union of the cities' geometries is in the department geometry. To aggregate the area of agricultural plots (spatial numeric measure Agricultural_plots_surface) of each city to obtain the total agricultural plots area for the department, one will simply add the values recorded for each city in the Puy-De-Dôme. We note that this is not always the case: for instance little isles belonging to a department could be left out of the department geometry during a cartographic generalization for performance purposes; the isles still belong semantically to the department but their geometries are not in the department geometry; aggregation in such cases can be pre-calculated based on the semantic relationships.

When the topological relationship is an overlap it is required to single out which measures (hold by a specific child member) can be considered entirely in the aggregation and which ones should be split relatively to the part of the child member geometry participating in the aggregation. For example, to aggregate the measure Agricultural_plots_surface for a watershed, one will have to define a specific procedure to take into account agricultural plots overlapping the watershed. These plots will be split and

only the plot parts that belong to the watershed will be taken into account, plus the whole surface of the ones which are in the watershed.

2.3.3 The SOLAP architecture

2.3.3.1 Functional architecture

SOLAP systems can be implemented in different architectures depending on the needs and constraints of the organization (Kimball and Ross 2002). We have chosen to present the most commonly implemented SOLAP architecture since it allows us to identify the essential SOLAP systems components (database, SOLAP server, client module) for our thesis.

It is a 3-tiers architecture, where useful detailed data are extracted directly from heterogeneous transactional sources available in a Resources tier. They are cleaned and transformed according to the analysis requirements, and then loaded in the SOLAP datacubes stored in spatial data warehouses (SDW tiers) during an Extract Transform and Loading (ETL) process. The SOLAP datacubes are interrogated by means of SOLAP operators executed at a SOLAP server tiers via a SOLAP client tiers. Those aggregation functions are exploited in SOLAP operators (or specific MDX queries) to interactively and easily analyze and explore data. There are various SOLAP operators including the spatial Slice, which selects a subset of spatial data; the spatial Roll-Up, which allows climbing into spatial hierarchy aggregating measures; and the spatial Drill-Down, which is the inverse of the Roll-Up.

In most cases, SOLAP datacubes are stored as tables in a relational database; it is the Relational OLAP (ROLAP) architecture (see Figure 2-4). To do so, the database is structured after one of the specific models that are the star model (Adamson 2006), the snowflake model (Jarke, Lenzerini et al. 2003), etc. A mapping is done between the data structure and the client multidimensional view (via an XML file for example) in the ROLAP server. That allows the ROLAP server to extract the data according to a multidimensional view before presenting it to the SOLAP client.

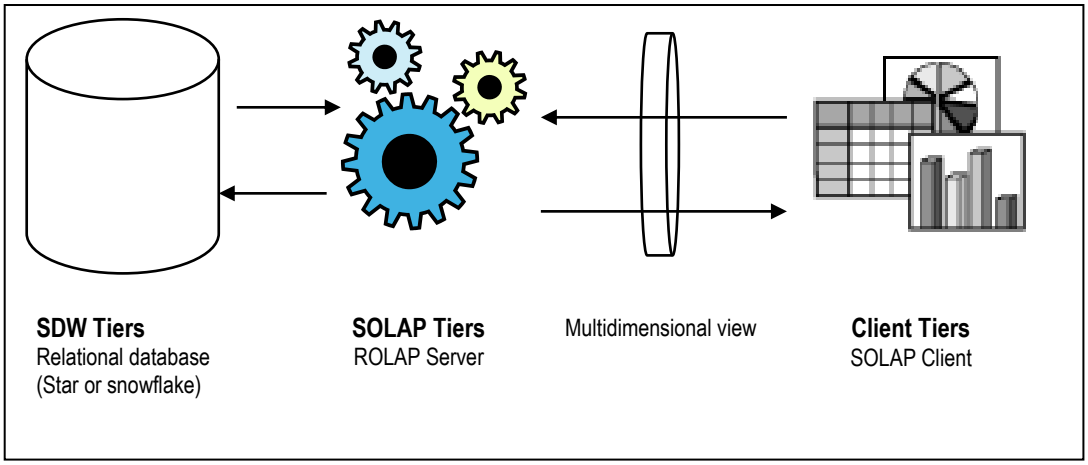


Figure 2-4 : ROLAP Architecture (adapted from Course SCG-7006[3] lecture notes)

In other cases, the SOLAP datacubes are stored in multidimensional databases (Multidimensional OLAP architecture - MOLAP) or in a combination of multidimensional and relational databases (Hybrid OLAP - HOLAP) (Salka 1998).

As for the SOLAP Client tiers, it combines and synchronizes tabular, graphical and map visualization of the data that allow end-users visualizing the data and triggering the SOLAP queries.

2.3.3.2 Technical architecture of a ROLAP system

SDW Tiers:

In the ROLAP system, the useful detailed data are extracted directly from heterogeneous transactional sources (e.g. Spatial databases stored in PostGIS/PostgreSQL, Oracle, Informix, ArcGIS, etc., data files), then they are cleaned, transformed if needed and loaded in the spatial data warehouse in a transactional system as well but in the form of a star schema, a snowflake schema, a mixed schema or a fact-constellation schema.

[3] Notions avancées de base de données SIG

The star schema (see Figure 2-5) is a well-known and commonly used data structure where the central element is the facts table (containing the facts) which is connected to dimension tables. Each dimension table contains the attributes holding its members.

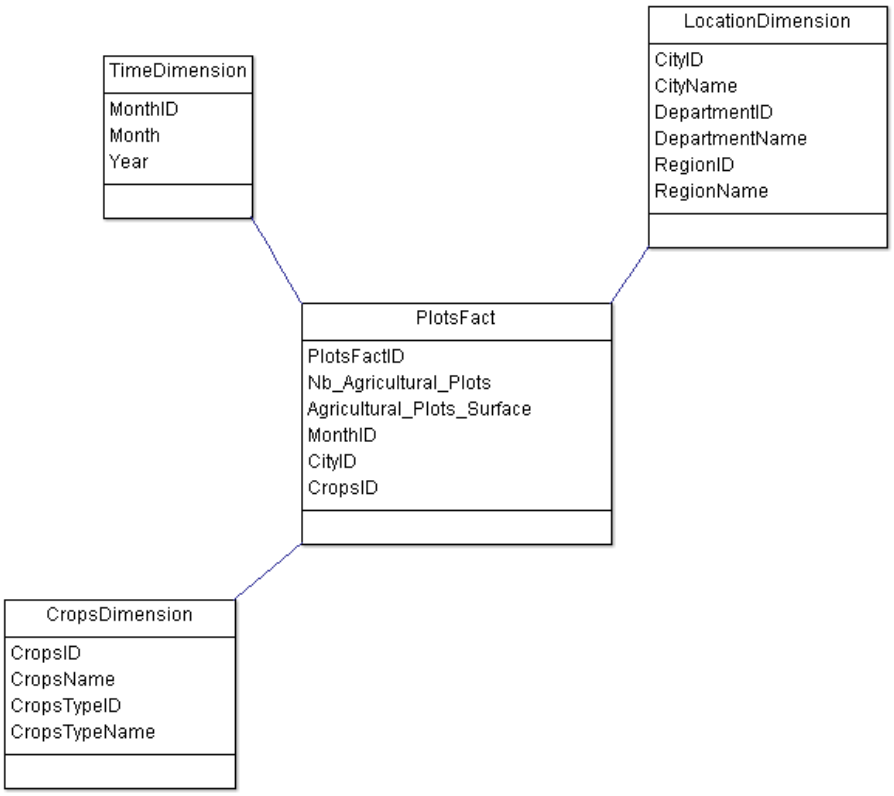


Figure 2-5 : Example of a star schema

The snowflake schema is a refinement of the star schema with each dimension being materialized in normalized tables containing the different hierarchies. The combination of the snowflake and star schemas result in the mixed schemas where only some dimensions (big tables) are materialized in many hierarchies tables. Finally, the fact-constellation schema is composed with many schemas (star of snowflake) where the fact tables share common dimension tables.

SOLAP Server Tiers:

There are more than fourteen (40) open sources (free solutions) or commercial SOLAP servers (Multidimensional or Relational) that are used in practice (Bédard, Proulx et al. 2005): Mondrian/GeoMondrian (open source and freeware), SQL Server Analysis Services and Oracle Data Analysis Oracle Spatial BI Enterprise Edition (commercial), Map4Decision server etc. They are all designed to manage crisp geometries. In this thesis, we will be using the Mondrian as SOLAP server because the Irstea team is already familiar with it.

Mondrian requires an XML file defining the SOLAP datacube schema with its dimensions and measures. It allows the connection to any DBMS in order to retrieve the data according to the definition provided in the XML schema. The company Spatialytics have developed the GeoMondrian, which is Mondrian extended with spatial components. GeoMondrian allows the definition of geometric spatial levels (members have vector geometries) and the execution of MDX queries integrating spatial predicates (union, intersection etc.). To the best of our knowledge, it only supports PostgreSQL/PostGIS as SDBMS.

Client Tiers:

For the Client Tiers, there are solutions such as Jrubik/JPivot (open source and freeware), Map4Decision client (shown in Figure 2-6) and SAS Web OLAP Viewer (commercial) etc. It is difficult to find a free client module that can be used with the GeoMondrian server in practice.

In this thesis, we will be using JRubik as client for our implementations because it offers the possibility to play with the visual parameter (such as cell colors).

JRubik, written in Java, allows the display of the datacube analysis results in pivot tables and histograms. It has a graphical interface allowing end-users to construct their queries by simple drag and drops and it also presents a MDX queries editor for advanced end-users. JPivot is the web version of JRubik and it presents the same features.

JRubik and JPivot solutions are well integrated with the Mondrian server but the spatial component is not well managed. Indeed Mondrian does not offer SOLAP operators (spatial drill-down, roll-up, slice etc.) and the map visualization and exploration in the two clients is very weak. Implementing a real SOLAP system using free solutions is therefore a mission impossible unless work is also done on the client module development (Bédard, Proulx et al. 2005, Malinowski 2014).

We note that classical commercial servers and clients are also designed to manage crisp geometries and not spatial vague objects models.

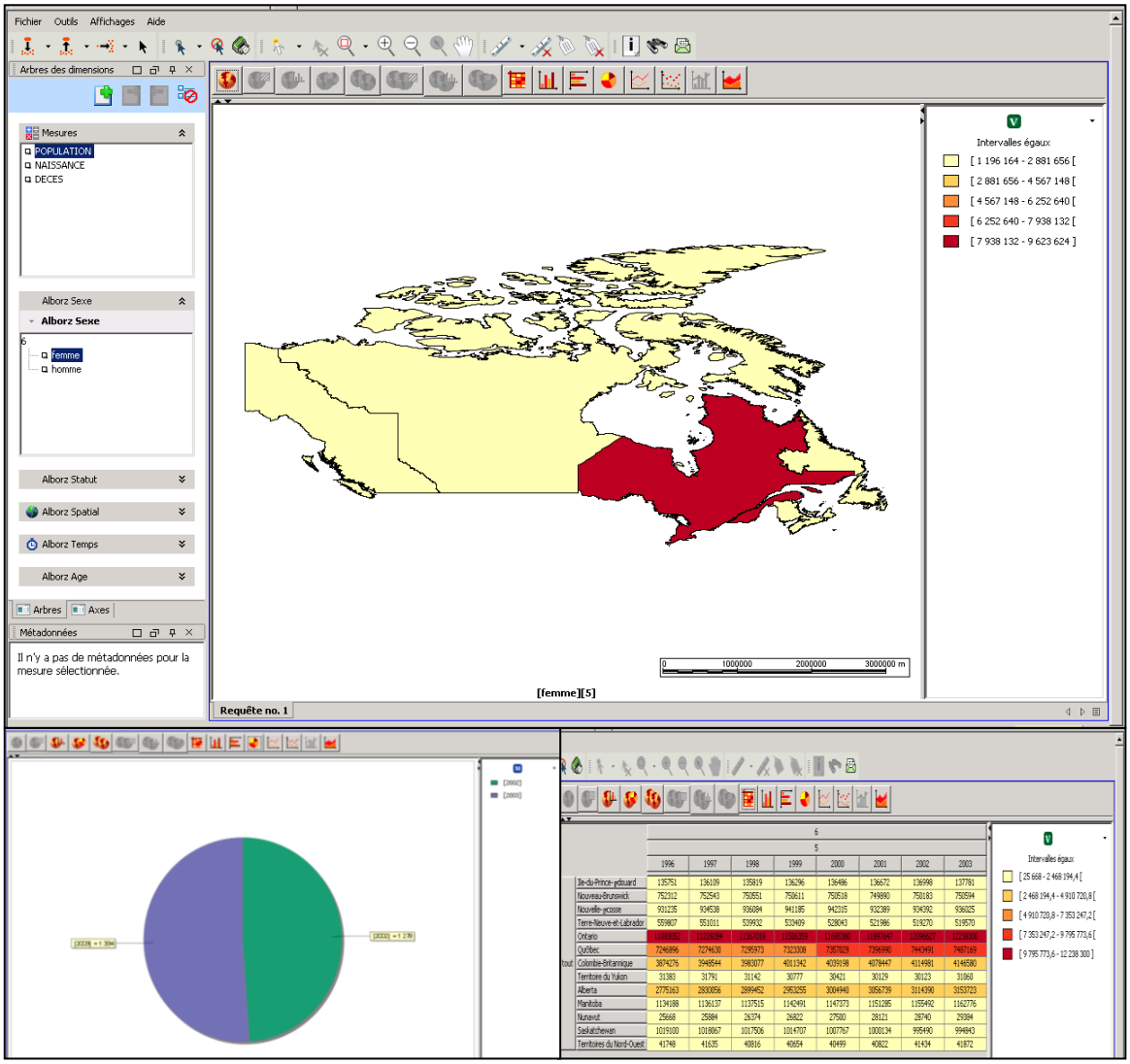


Figure 2-6 : Map, diagram and pivot table visualization in a SOLAP client (Map4Decision)

2.3.4 SOLAP datacubes production

As shown in the previous sub-section (section 2.3.3) SOLAP systems are based on multi-tiers architectures where (spatial) data should be extracted from heterogeneous data sources and transformed before to be loaded into the datacubes. This process (ETL) is very time and money consuming (Bédard 2007). Indeed, during the ETL phase, producers must not only make sure that the data sources are topologically correct and coherent regarding the updates, and that the objects geometry is appropriate for each level, but also that the data uncertainty is well managed (Bédard, Rivest et al. 2006).

Moreover, since SOLAP datacube model represents required SOLAP analyzes, the success of the whole SOLAP application project depends on the correctness of the designed SOLAP datacubes.

The following paragraphs introduce us to both SOLAP application elaboration methods in general and SOLAP datacube design approaches in particular, as advocated in the literature.

A SOLAP application project is structured in four main phases (Figure 2-7): the requirements specification phase where end-users needs are analyzed, the conceptual-design phase where conceptual SOLAP datacube schemas are produced, the implementation phase where SOLAP datacube logical and physical schemas are produced and the feeding phase where ETL operations are conducted (Malinowski and Zimányi 2008). The project can be executed by following methods such as Model Driven Architecture-based (MDA) methods as presented in Glorio and Trujillo (2008) and/or a rapid prototyping ones (Guimond 2005, Bimonte, Nazih et al. 2013).

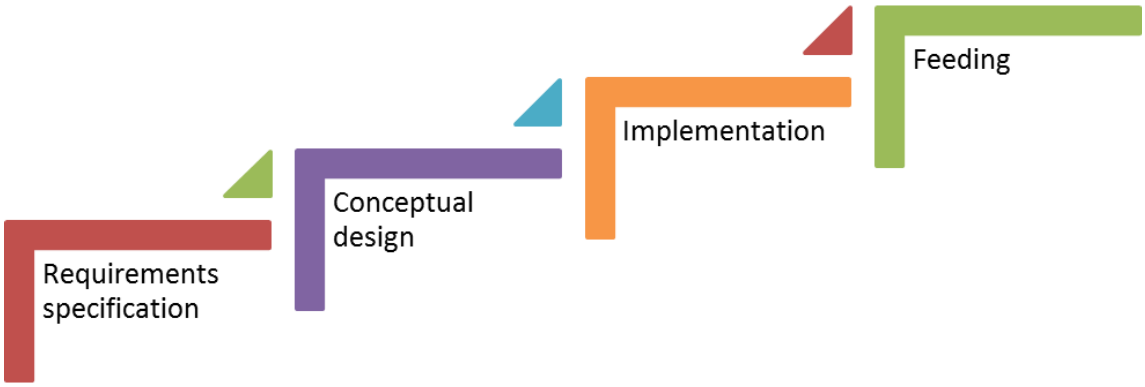


Figure 2-7 : SOLAP application project phases

The MDA method is an Object Management Group standard that advocates the use of models in software development (OMG 2003). It addresses the complete life cycle (design, implementation and management) of systems by providing an approach and tools for:

- Specifying a system Computation Independent Model (CIM) to simply describe the domain, the situation in which the system will be used without showing any information about how it will be structured. The CIM is the absolute conceptual level because it does not involve the implementation technology. Regarding SOLAP applications, this type of model can be elaborated during the requirements specification phase to support the communication between SOLAP experts and end-users (Glorio and Trujillo 2008).

- Specifying a system Platform Independent Model (PIM) that describes the system structure according to a specific family of technologies (SOLAP systems, transactional systems etc.), but independently of any particular platform that will support it (Oracle, PostgreSQL/PostGIS etc. in the transactional systems family). The PIM is seen as a conceptual model relatively to a specific platform since it does not describe the actual implementation (Glorio and Trujillo

2008). This type of model is typically elaborated during the conceptual design phase for SOLAP applications.

- Specifying a system Platform Specific Model (PSM) by transforming the PIM with addition of pertinent information needed for the implementation on the chosen platform (Oracle PSM, GeoMondrian PSM etc.). This type of model can be elaborated during the implementation phase for SOLAP applications.
- Transforming the system specification (PSM) into physical models (implementation code in SQL, XML etc.) adapted for the platform chosen. It is done also in the implementation phase for SOLAP applications.

Although the models can be elaborated using any modeling language, the Unified Modeling Language (UML) is widely adopted because MDA and UML are related historically and in practice. Indeed, the OMG members have chosen, since 2001, a unified approach that advocates the exploitation of the MDA method with a modeling language based on the MetaObject Facility (MOF), which is actually built on UML 2.0 by the same organization (OMG 2011). Using UML for the models is thus coherent with the usage of UML for the meta-models because it fits into the OMG unified approach. Also the fact that UML is a well-known standard that can be adapted to particular domains via the enrichment of meta-models or profiles (definition of new tagged values and new classes and attributes stereotypes) contributes to its popularity.

Few years ago, to help guarantee the SOLAP application usability and end-users satisfaction, Guimond (2005) had encouraged the use of prototyping to quickly discover and specify end-users needs in analysis. It had been noticed from experience that end-users did not realize the full capacity of SOLAP which make it really difficult for them to express their needs. Thus, with the prototyping approach, they were provided very quickly with prototypes they could “play” with to properly and fully express their needs and/or validate the design. The highlighted principles in this context are iteration, rapid prototyping and user involvement. Now, it has become simpler, easier and more productive to directly elaborate functional classical SOLAP applications with Map4Decision than to prototype them first. However, the key point here is that rapid prototyping is proven to be efficient when end-users need to be provided very quickly with applications but the regular process does not allow it.

The SOLAP datacube design itself is done during the first two phases which are the end-users requirements specification and the conceptual design phase. In the literature, several works investigate OLAP/SOLAP datacubes design (Giorgini, Rizzi et al. 2005, Guimond 2005, Prat, Akoka et al. 2006, Malinowski and Zimányi 2008, Romero and Abelló 2009, Pardillo, Mazón et al. 2010, Romero and Abelló 2011, Di Tria, Lefons et al. 2012). They provide guidelines, processes, methods and/or tools that can be grouped into three main categories according to the approach used for specifying the multidimensional model: the *users-driven* methods, where a multidimensional intelligence is drawn from the requirements expressed by end-users; the *sources-driven* methods, where a multidimensional intelligence is (manually

or semi-automatically) drawn from identified data sources; *hybrid* methods that exploit both sources data and end-users requirements in specifying the multidimensional model (*de-facto* used approach for SOLAP systems in particular).

In the computer science community, in general, the user-driven and data-driven steps are conducted in parallel during the requirements specification phase with as results two PIMs. The two models are then transformed into two CIMs and then matched during the conceptual design phase to produce a unique CIM for the intended OLAP datacube (Cf. Figure 2-8). Some researchers have advocated techniques and approaches for the models matching (Bonifati, Cattaneo et al. 2001, Mazón, Trujillo et al. 2007) while others have proposed a hybrid approach where the user-driven and data-driven activities are done simultaneously (user-driven requirements are exploited to reduce the scope of possible multidimensional factual data extracted from the sources or vice versa) (Phipps and Davis 2002, Winter and Strauch 2003, Malinowski and Zimányi 2008, Romero and Abelló 2010).

In the Geomatics community however, first end-users' are identified to analyze the domain and their needs. In the meantime, the available data-sources are also identified and, guided by the end-users and their needs, multidimensional knowledge (measures, dimensions, etc.) is extracted from the sources. That knowledge will enrich the overall analysis requirements specification. In the conceptual design phase, the SOLAP datacube conceptual schema (PIM) is developed according to the requirements. We have translated the process described in this paragraph in an UML activity diagram that can be seen on Figure 2-9 for a better comparison with the OLAP hybrid design process.

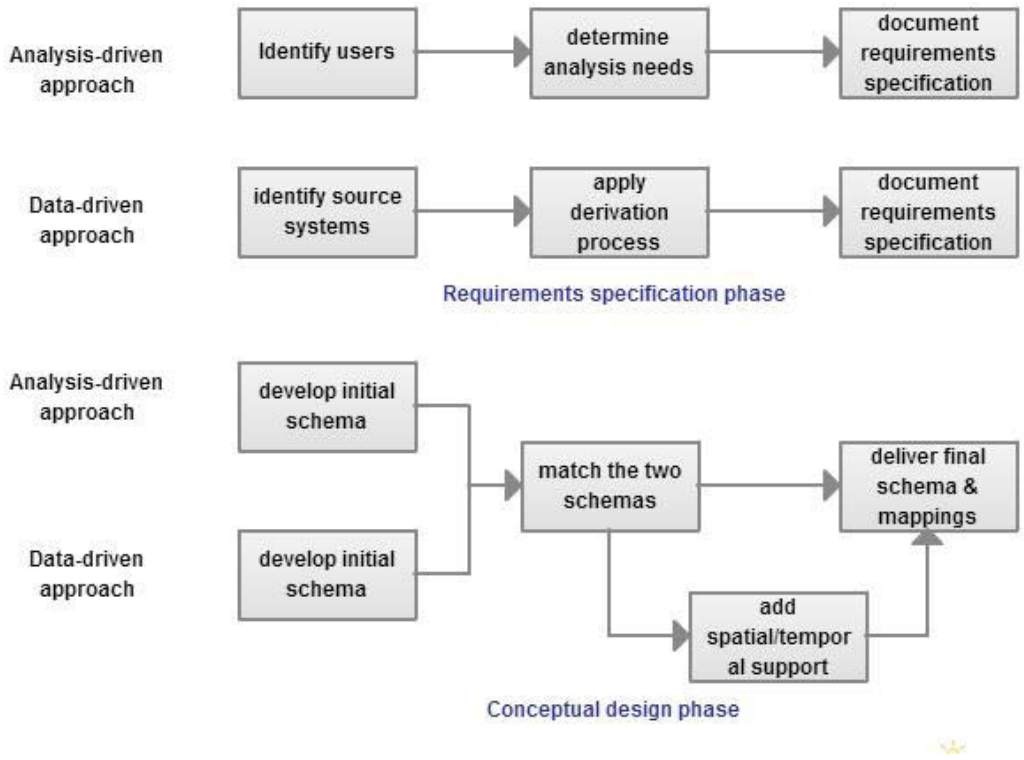


Figure 2-8: OLAP datacubes hybrid design approach – adapted from (Malinowski and Zimányi 2008)

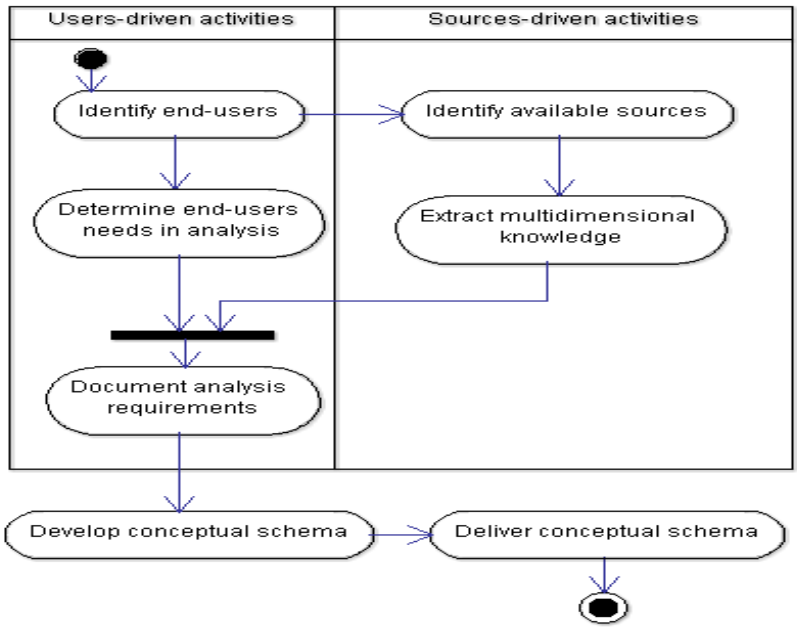


Figure 2-9: SOLAP datacubes hybrid design approach

2.4 Risks of misuse: definition and management

Risk management is a great topic of interest for project management and natural risks domains as well as the ISO regarding the usage of products, processes and services. The risk management refers to the reduction of a risk to a level considered acceptable (Gervais, Bédard et al. 2009). In the recent years, the management of the risk of spatial data misuses has become one of the main preoccupations and thus the subject of a growing interest from the Geomatics community (Lévesque 2008, Gervais, Bédard et al. 2009, Gervais, Bédard et al. 2012, Grira, Bédard et al. 2013, Roy 2013). This is because it has become clear that managing the risks of misuse is one of the best ways to reduce uncertainty and prevent inappropriate use of geospatial data by all type of users (expert or non-expert of the spatial data). Our objective here is to understand the concept of risk and risk management enough to be able to address risk-awareness in design methods in this thesis. Therefore, this sub-section will focus on the risk management definition and management translated to the Geomatics world, and more specifically to SOLAP datacubes.

In their recent work Lévesque (2008) and Gervais, Bédard et al. (2009) specifically define the ***risk of SOLAP datacube misuses***, by extending the ISO/IEC-51 (1999) definition, as the ***probability of occurrence of an inappropriate use of SOLAP datacube combined with the severity of the inappropriate use***. It is represented by a qualitative value (e.g. low, medium, high), which varies according to the context of use of the datacube.

Risk management is a topic that has been addressed in project management and by the ISO (ISO/IEC-51 1999) as stated before. The approaches proposed are more or less similar: in general, it is recommended to properly identify and describe the potential risks related to a project or a usage at first. That way, it is possible to evaluate those risks and really identify appropriate strategies to reduce the risks and thus prevent the risks consequences. Lévesque (2008) and Gervais, Bédard et al. (2009) have adapted the risk management method proposed by the and in project management field to the geospatial domain and more specifically to SOLAP datacubes and their risk of misuse. The results of their work is presented on Figure 2-10. It is a risk management method which takes place in five main phases:

(1) Identification of possible inappropriate usages of the SOLAP datacube (risks identification): This step is important because it is where the risks to be managed are defined; scenarios of inappropriate use must be identified knowing the intended usage context, the data source quality (through metadata, data dictionaries, data sources etc.), the processes applied or to be applied on the data etc. The more exhaustive the list is, the more successful the management can be.

(2) Analysis of the risks (***risk analysis***): Assessing the identified risks is not an easy task but it is also an important one in the risk management process. The risks probability of occurrence and severity levels must be assessed, using among others a risk degree scale (Low, Medium, and

High for instance). The analysis require a good knowledge of the usage context as well as an experience and a good understanding of the spatial data involved.

(3) Evaluation of the risks (***risk evaluation***): Usually, the risk evaluation is done by combining the risk severity and probability degrees using a risk degrees matrix. On

(4) Table 2-1 we present the one proposed by Kerzner (2006) in the context of project management. It is a matrix where the axes are respectively the risk probability of occurrence and its severity (or gravity). The intersections correspond to the risk global dangerousness, which can be High (H), Medium (M) or Low (L). For example, for a risk with a very high probability of occurrence (E) and very low consequence gravity (A), the overall dangerousness is set at medium.

Consequence ⟶ Higher

Higher ↑ Probability

	A	**B**	**C**	**D**	**E**
E	M	M	H	H	H
D	L	M	M	H	H
C	L	L	M	M	H
B	L	L	L	M	M
A	L	L	L	L	M

Table 2-1: Risk evaluation matrix (Kerzner 2006)

(5) ***Risk response***: Here, the method recommend to choose different actions/strategies to deal with the identified risks. The response is tailored to the risk degree but also to data producers and end-users. Indeed, depending on the risk probability of occurrence and gravity, SOLAP datacube producers can decide to ignore the risk and deliver the datacube as such, or they can choose to reduce it by eliminating its source (avoidance strategy), transferring the risk to a third party (end-users or by contracting an insurance), or controlling the risk the risk control by taking preventive actions. In practice, data producers or users can control the risks of misuses related to spatial data in different ways. For example, according to a Canadian survey presented in Gervais, Bédard et al. (2012) work, some producers choose to limit the access to their data to informed users while others prefer delivering a list of recommended and non-recommended usages with the data.

(6) Risk monitoring and audit (***risk audit***): This is where the risks are documented to allow the monitoring and experiences capitalizations for future risk management activities. It is a step that

was specifically added to the ISO risk management method by the authors for legal purposes and because it helps for future SOLAP datacubes elaborations.

Finally, if there are still risks that are not tolerable, the whole process should be started over. The method is iterative.

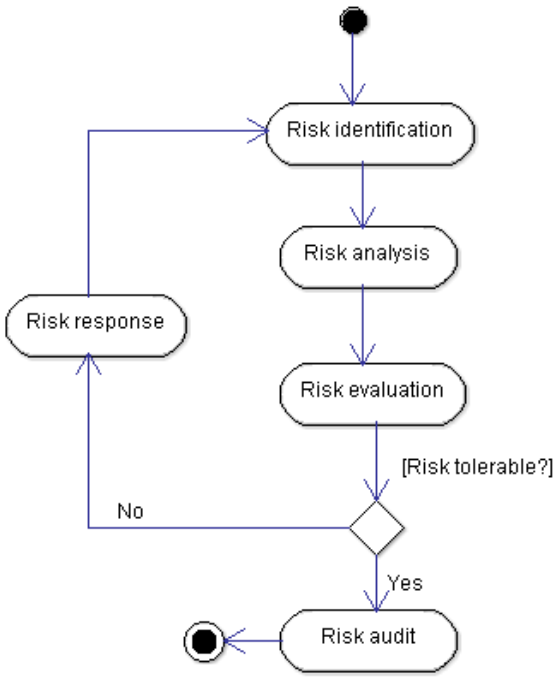


Figure 2-10: Risk of misuse management method: adapted from Gervais, Bédard et al. (2009)

2.5 Chapter synthesis

In this chapter, we described fundamentals concepts related to the topics covered by this thesis with the goal of getting a better understanding those key concepts. At first we described and defined spatial data uncertainty, spatial vagueness and spatial data quality. Then we present an overview of SOLAP systems concepts, from the multidimensional model to SOLAP datacube production with an emphasis on the ROLAP architecture and existing SOLAP tools. Finally, we present the concept of risk of misuse and described the risk of misuse management method that will be at the basis of the risk-awareness proposals in our work.

Chapter 3: New Design Approach to Handle Spatial Vagueness in Spatial OLAP Datacubes: Application to Agri-environmental Data

Accepted in International Journal of Agricultural and Environmental Information Systems (IJAEIS), 2015
Edoh-Alove, E., Bimonte S., Pinet, F., Bédard, Y.

3.1 Introduction

This chapter addresses the first sub-objective of our thesis which is to propose the fundamentals of a new risk-aware SOLAP datacubes design approach.

As stated previously in Chapter 1, spatial data always suffer from different levels of uncertainty. Not dealing with uncertainty, especially the spatial vagueness, when making high-level decisions based on SOLAP aggregated data increases the risks of data misinterpretations (Gervais, Bédard et al. 2009). This leads to faulty trend analysis, missed problems and inexact comparisons between regions or periods. To deal with spatial vagueness in SOLAP systems, two main approaches are investigated in the literature. The first one tries to reduce the uncertainty (overabundance of observations to increase spatial precision for example) from the data or to provide decision-makers with visual feedbacks about the uncertainty (Worboys 1998, Lévesque 2008, Bimonte, Nazih et al. 2013). The second one proposes to handle the uncertainty issues by using new uncertainty-aware spatio-multidimensional models and operators (Perez, Somodevilla et al. 2007, Jadidi, Mostafavi et al. 2012, Siqueira, Aguiar Ciferri et al. 2012), that are based on the representation of the vague objects with fuzzy or exact models. Not only the implementation of those solutions is still in an embryonic state, but also the new paradigms brought by the uncertainty-awareness in datacubes make the datacubes implementation and analysis more complex for designers and end-users respectively. Motivated by the desire to offer a solution that presents a symbiotic trade-off between the theoretical accuracy on spatial vagueness, the implementation feasibility in current technologies and the usability by intended end-users, we come up with a third approach: **instead of dealing with the complexity of manipulating complex vague objects models in SOLAP systems, we propose to manage the risks of SOLAP datacubes misinterpretations, related to spatial vagueness, that the end-users incur.** To do so, we define a new SOLAP datacubes design approach that can take those risks into account during the datacubes modeling process. Such approach leads to the development

of a classical SOLAP datacube which not only fits the end-users' usage, but can also be implemented in existing (commercial) SOLAP tools and explored with classical SOLAP operators.

This new approach extends existing methodologies with three main elements. First, it takes simultaneously into account available data sources, end-users' needs and end-users' tolerance levels to «well-identified risks of SOLAP datacubes misinterpretations due to spatial vagueness issues». Second, it delivers to end-users, different versions of SOLAP datacubes, one for a set of tolerance level, where the possibility of making erroneous SOLAP analyses is minimized. Third, it enriches the SOLAP datacubes elements with visualization policies to properly communicate risks of misinterpretations to end-users if necessary.

In this chapter, at first we provide the definition and classification of the risks of SOLAP datacubes misinterpretation induced by the presence of vague spatial data in the datacubes. We then detail our proposal of a new risk-aware design process that aim to help datacubes producers design datacubes while considering the vagueness and the risks associated. The users' tolerance levels scale and possible risk management strategies and actions that support the new approach are presented here as well. We also offer an illustration of the approach on agri-environmental data.

These results have been compiled, into a scientific paper that has been submitted and accepted in the double-peer reviewed journal International Journal of Agricultural and Environmental Information Systems (IJAEIS) in July 2014.

The chapter is organized as follows: Section 3.2 presents a the state-of-the-art on spatial vagueness management in SOLAP systems; motivation of our work using an agricultural case study is presented in Section 3.3; in section 3.4, we define and classify the risk of misinterpretation before moving on to defining our new risk-aware design approach requirements as well as the whole new design process proposed in section 3.5; in section 3.6 we detailed our contributions regarding the risk of misinterpretation assessment and management in the new approach; finally the approach is tested on the case study in section 3.7.

3.2 Literature review

There are several strategies for managing spatial data uncertainty in general (Gervais, Bédard et al. 2007): provide the data without any indications or treating the uncertainty; model the uncertainty and communicate it to users; reduce the uncertainty ; make the data producer, user or both absorb the uncertainty or transfer it to a third party (the producer can take an insurance to cover users who suffered damage because of their data) (Bédard 1986).

In order to reduce uncertainty related to spatial vagueness, researchers, throughout the years, have focused on the use of vague objects models (as opposed to crisp objects types point, line, polygon) to

represent some spatial phenomena in a more accurate way. Thus, four categories of vague objects models have been advocated: exacts models, fuzzy models, probabilistic models and rough models.

The fuzzy model is based on the Fuzzy Set theory (Zadeh 1965) which describes the possibility that an individual is a member of a set or that a statement is true. Given points of a set have a membership degree (in the interval [0,1]) computed using a membership function. The fuzzy logic is opposed to the Boolean logic where an individual belonging to a set is evaluated at false (non-member of the set) or true (member of the set). In the fuzzy context, the spatial object with vague shape or location is generally described as a continuous fuzzy set associated with a membership function (monotonous, bell-shaped, or triangular function etc.). The membership function definition depends on the phenomenon modeled and it is the most difficult part of the fuzzy modeling.

In exact models, the geographic information is represented by a complex geometry consisting of at least two crisp geometries (Cohn and Gotts 1996, Bejaoui 2009, Pauly and Schneider 2010): one represents the minimal extent/core of the phenomenon (area where the phenomenon is surely present) , and the other represents its maximal extent/dubiety (area where the phenomenon is probably present). The exact model is designed to make use of traditional crisp concepts and in consequence to be implementable in classical systems. Other alternatives are to represent the information with rough sets (Worboys 1998) where the spatial object is described as an approximation classification with the maximum approximation reflecting the uncertain part of the modeled object; and probabilistic models (Burrough and Frank 1996).

In the following paragraphs, we present the fundamental concept of the Qualitative-Min-Max model proposed by (Bejaoui 2009) in our research group, which allows the representation of vague spatial objects by means of crisp geometries (exact model).

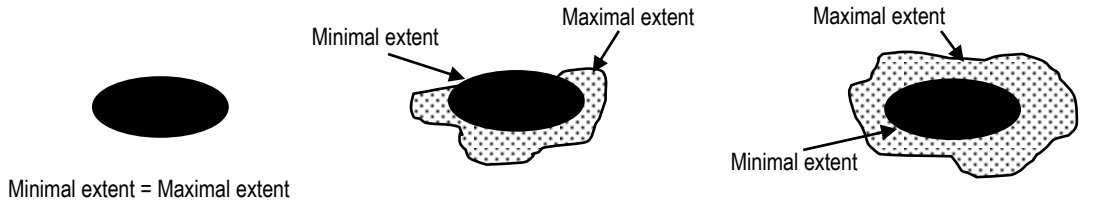
The QMM model defines a region with broad boundary as being a region composed of two simple crisp regions: a *minimal extent* and a *maximal extent*.

The *minimal extent* is the “representation of the region when the boundary is considered as close as possible”. It is actually the set of points certainly belonging to the region with broad boundary.

The *maximal extent* is “the representation of the region when the boundary is considered as far as possible”. It is the union of the minimal extent and the set of points possibly belonging to the region with broad boundary.

The *maximal extent* thus defined may contain, cover or be equal to the minimal extent. When the maximal extent is equal to the *minimal extent*, the region is a crisp region or a region with no broad boundary. The model also distinguishes between a region with partially broad boundary (*maximal ext*ent is equal to

minimal extent only at some parts of the region) and a region with completely broad boundary (Cf. Figure 3-1).



(a) A crisp region *(b) A region with a partially broad boundary* *(c) A region with a completely broad boundary*

Figure 3-1 : Regions with broad boundaries (Bejaoui 2009)

This concept of region with a broad boundary includes the concept of region with no broad boundary. It can be exploited to specify a more accurate representation of phenomena with vague shapes and location by means of crisp geometries. To do so, it is necessary to associate with each geographic object at least two polygons representing the different extents. Such representation calls for the definition and control of the topological relationships between the geographic objects not only for their storing but also for their analysis. In this regard, (Bejaoui 2009) has proposed an adverbial Spatial Objects Constraint Language (Spatial OCL[4]) for defining topological integrity constraints for objects with vague shapes. Note that to present, the majority of GIS and spatial databases still manage only simple crisp objects types (Pauly and Schneider 2010).

Spatial vagueness management in SOLAP systems

Very little researches focus on the integration of vague objects models in SOLAP systems. The authors Siqueira, Aguiar Ciferri et al. (2012) have recently proposed an extension of the multidimensional model for taking into account the exacts models of Bejaoui (2009) and Pauly and Schneider (2010) with specific techniques for storing and querying (vague window query) the vague SOLAP datacubes (e.g. storing separately the core and the conjecture region of a vague object). Yet, they do not offer tools based on a practical implementation of their new definitions and techniques. Jadidi, Mostafavi et al. (2012) propose an algorithmic approach based on fuzzy set theory to integrate vagueness in the decision making process. Concretely, they work out a fuzzy spatial model (fuzzy spatial dimension, spatial fact, spatial measure etc.), a fuzzy spatial aggregation model (fuzzy union, intersection, overlay, etc.) and fuzzy indicators to manage and analyze areas at risk of coastal erosion in SOLAP datacubes. The implementation of the

[4] Extension of the OCL as advocated by Duboisset, M., F. Pinet, M. A. Kang and M. Schneider (2005). "Precise modeling and verification of topological integrity constraints in spatial databases: from an expressive power study to code generation principles." Lecture Notes in Computer Science (ER) **3716**: 465-482.

fuzzy spatial datacube is identical to classic datacubes except that we have to add an attribute to support the membership values with regards to the spatial members. Nevertheless, the measures should be calculated with the fuzzy operators instead of classic ones. The implementation of the defined operators in commercial SOLAP clients is yet to be done, and also, the computation of the membership functions requires specific practical tools that are not necessarily available.

As far as we know, research on spatial vagueness introduction in SOLAP datacube visualization is non-existent for now, although SOLAP has been used to visualize spatial vagueness in transactional geospatial data (Devillers, Bédard et al. 2005, Devillers, Bedard et al. 2007).

In the literature related to the quality of OLAP systems, there have also been proposals dealing with data fuzziness in particular. Some of the authors in this field advocated the use of fuzzy set theory (Laurent 2002, Delgado, Molina et al. 2004, Fasel and Zumstein 2009) or rough set theory (Naouali and Missaoui 2005, Naouali and Missaoui 2006) to handle the data incompleteness and imprecision. The first ones proposed fuzzy multidimensional database models and fuzzy slices, dices and aggregations while Naouali and Missaoui (2005) and Naouali and Missaoui (2006) contributions involve data approximation techniques by means of rough sets (upper and lower approximations). Also, the authors Pitarch, Favre et al. (2012) exploited the fuzzy logic in proposing new contextual hierarchies to allow the consideration of data experts knowledge for the definition of the hierarchies.

Although it is interesting to integrate vague objects models in SOLAP datacubes, there is still much to do before we can design, implement and exploit these vague datacubes in practice. Existing SOLAP tools (Extract Transform Load tools -ETL, Servers, Clients etc.) and databases (Pauly and Schneider 2010) only allow the storage and querying of spatial data modeled as crisp entities (point, line, and polygon).

(S) OLAP datacubes design methods

OLAP and Spatial OLAP systems are based on complex multi-tiers architectures where (spatial) data should be extracted from heterogeneous data sources and transformed before being loaded into the datacubes. Since SOLAP datacube multidimensional model represents required SOLAP analyses, the success of the whole SOLAP application project depends upon the correctness of the designed SOLAP datacube(s). In the previous years, several multidimensional modeling methodologies where presented in the literature (Giorgini, Rizzi et al. 2005, Guimond 2005, Prat, Akoka et al. 2006, Malinowski and Zimányi 2008, Romero and Abelló 2009, Pardillo, Mazón et al. 2010, Romero and Abelló 2011, Di Tria, Lefons et al. 2012). They offer guidelines, processes and/or tools to designers to help them draw (manually or automatically) a multidimensional intelligence from end-users' requirements (requirement-driven approach) or data sources (sources-driven approach) or a combination of both (hybrid approach; de-facto used for SOLAP systems). It is implemented in a Model Driven Architecture (MDA) method (Glorio and Trujillo 2008) and/or a rapid prototyping one (Guimond 2005). The analysis of these methodologies shows that

(spatial) data uncertainty issues do not explicitly influence the resulting datacube multidimensional elements definition.

Indeed, on one hand, user-driven methods focus on determining and exploiting users' needs in defining the multidimensional elements. They are not designed to identify and/or consider the data types or uncertainty issues regarding the data to be exploited in the datacubes. On the other hand, the sources-driven methods allow the inventory of attributes present in the sources and their types so that the measures and dimensions can be extracted from the sources; but the methods do not allow considering the attributes content. However, knowing that an attribute holds a spatial vague object is needed to consider spatial vagueness issues soon enough during the design process. Ultimately, most hybrid methods present the same limits since they usually focus on resolving the merging of user-driven and sources-driven results (Bonifati, Cattaneo et al. 2001, Mazón, Trujillo et al. 2007). The methods used in practice in Geomatics, advocate the analysis of the sources and their quality according to the needs but in those methods, the quality issues (spatial accuracy, incompleteness, generalization issues (Bédard, Proulx et al. 2005) etc.) are whether resolved during the ETL process, or during the datacube visualization and exploration. Recently, approaches have been proposed to integrate a risk of misuse management method to the design methods, allowing the producers to identify and assess those risks early on during the design. It is for example the proposal of a collaborative platform for the risks identification and analysis by Grira (2014); or the work of Lévesque (2008) leading to the proposal of risks of misuse classification and identification formal tools, as well as contextual alerts definition allowing the display of the risks to the end-users during the datacubes exploration, to reduce the risks. Grira (2014) work is set on risk definition and assessment and can be seen as complementary to our proposal. Lévesque (2008) does not address spatial vagueness in particular and also the risks are not exploited in defining the final SOLAP datacubes elements.

3.3 Motivation

We consider a stripped down agri-environmental case study on pesticide spreading activities data. We consider the case where the intention is to build a SOLAP application for decision-makers to support their decisional process in the context of the control of surface water contamination by pesticide. The control activity implies the monitoring of the quantity of pesticide that can contaminate surface waters. To do so, we have on one hand (1) decision-makers analysis needs and on the other hand (2) the available source data.

(1) Regarding decision-makers analysis needs:

Decision-makers expect a SOLAP datacube that will help them visualize and interrogate data related to all pesticide spreading activities in an easy and interactive way. The datacube should allow answering queries such as:

(Q1) *"What is the **total quantity** of pesticide applied per year that can be found within flood risk areas?"*

(Q2) *"What is the **greatest amount** of pesticide applied per year that can be found within flood risk areas?"*

To help decision-makers monitor the quantity of pesticide in spread zones and flood risk areas, the required datacube will need to have a spatial hierarchy composed of at least a Spread Zones level (holding areas where pesticide have been applied), a Flood Zones level (holding flood risk areas), and also a measure that holds the quantity of pesticide spread values (QuantityAsub).

(2) Regarding the source data:

First, we note that spread zones (defined as areas where pesticide has been applied) and flood risk areas (defined as regions where surface water can be present during an inundation) are vague objects (see sample on Figure 3-2). More specifically, both spatial objects have vague shapes.

In fact, the spreading activities should be conducted on plots parts defined beforehand as suitable areas. Those areas are surely spread but due to the spreading equipment and technics, pesticide can be found outside those suitable areas. Thus, while spread zones maximal extents are the limits of farming plots (i.e., no pesticide are spread outside the farming plots boundaries), their minimal extents are the suitable areas. Even if the quantity of applied pesticide is exactly recorded for a whole plot, it is not possible to determine with accuracy where that quantity has been applied inside the limits of farming plots. However, in the sources, this vagueness has been neglected and spread zones are represented by polygons covering the whole farming plot.

Concerning the flood risk areas, official data sources provide well calculated geometries representing flood-prone areas. However, during an inundation, surface water does not always cover a whole flood-prone area. Therefore the limits of those areas are rather maximal extents of flood risk areas. Meanwhile, areas that are certainly covered by water during an inundation, meaning minimal extents of flood risk areas, correspond to the actual limits of water bodies such as rivers, lakes etc. plus a little buffer of 5 meters. The latter are stored in spreading activities database as unsuitable areas for pesticide spreading.

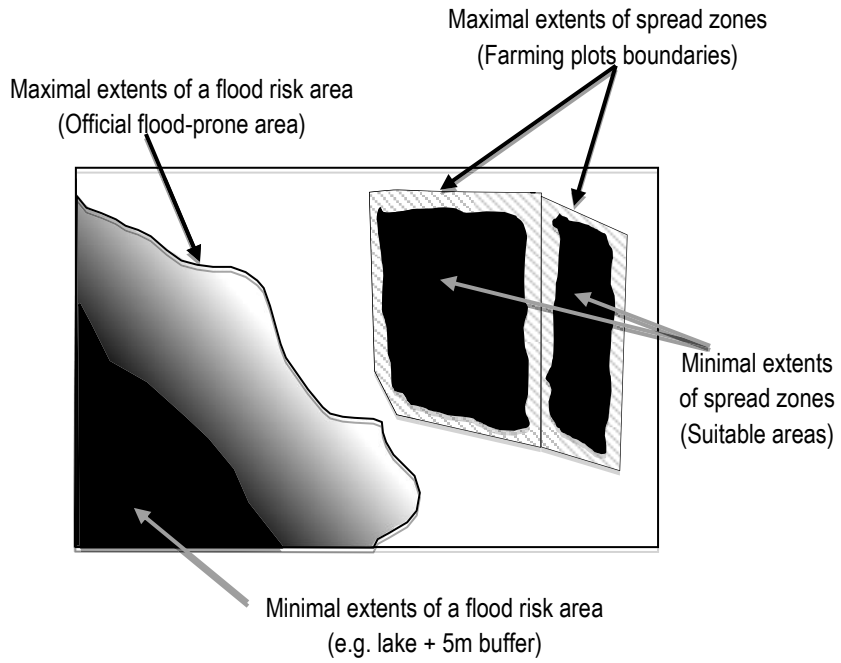


Figure 3-2: Map showing the vagueness on flood risk areas and spread zones

Farming plots are areas where agricultural operations are conducted; they are delimited by well-defined cadastral limits. Farms are polygons representing the union of farming plots that belong to the same farm operator. Both geographic objects are spatial vagueness free.

Exploiting the available source data (where spatial vagueness were neglected) to feed the expected datacube, decision-makers will be provided with a datacube where the spatial vagueness is also neglected. Therefore, the results of the queries will be uncertain (e.g. the analyzed value, as well as the members' geometry shown, for a level composed of spread zones), yet end-users may think they are accurate (risk of misinterpretation).

3.4 Risks of misinterpretation: definition and classification

In this section which launches our proposal, we answer two key questions related to our approach about the risk of misinterpretation: What does one call risk of misinterpretation? What type of risks of misinterpretation can one identify?

Starting from the risk of misuse definition proposed by Lévesque (2008) following the International Organization for Standardization (ISO) risk analysis standards (ISO/IEC-51 1999), the risk of datacube (measures and level attributes) misinterpretation can be defined as being *the result of combining the probability of occurrence of a datacube misinterpretation with the severity of the potential impact of the misinterpretation*. The risk of misinterpretation is represented by a qualitative value (e.g. low, medium, high), which varies according to the end-users' context of use since both the occurrence probability and impact depends upon this context.

A datacube (measures and level attributes) misinterpretation occurs when a decision-maker makes decisions from faulty information leading to unexpected results. Our interest is towards unexpected results when spatial vague objects are involved. Also in this thesis, we **focus on the numerical measures misinterpretation.**

We recall that in our symbiotic trade-off approach, datacubes geometric attributes are represented with simple polygons to allow an implementation in existing SOLAP systems. **Accordingly, to still take into account the vagueness on spatial data, the geometric attributes will contain either the minimal extent or the maximal extent of the vague objects (not both as it is done for the complex vague objects models in Siqueira, Aguiar Ciferri et al. (2012)).** For example, a flood zone can be considered only in its maximal extent (official flood-prone area), which will be represented by a single polygon, instead of being represented by a combination of two polygons (one corresponding to the official flood-prone area and the other to the lake+ 5m of buffer extent).

We classify the risks of measures misinterpretation in two main categories, which are **intrinsic risks** and **extrinsic risks**:

- **Intrinsic risks** are usage-independent risks. They are specific to the geometric data exploited in the datacube. They are first and for all potential risks, since a datacube end-user can still decide that their impact is negligible and that they do not represent a real risk for them depending on their usage.

For these intrinsic risks, we will use the term risks of poor measure evaluation (where “measure” is used in its datacube sense). The intrinsic risks can be:

  - Over evaluation;
  - Under evaluation;
  - Non-significant. There is nothing to report in this case; it happens for example when the vagueness is absorbed by the datacube aggregation structure.

- **Extrinsic risks** are usage-dependent risks. They are specific to the intended usage and depend upon the user context of application.

In this thesis we deal with the intrinsic risks. More specifically, we focus on the intrinsic risks related to the exploitation of spatial level vague geometric members. In other words, our interest is towards numerical measures poor evaluation that can occur when spatial levels of a spatial dimension contain vague geometric members (minimal extents or maximal extents, not both).

Intrinsic risks are concretely induced either by the vagueness on the spatial level geometric members (we call it "*Risk-Geometry*"), or by the aggregation algorithm (we call it "*Risk-Aggregation*").

(1) ***Risk-Geometry***: When the geometric members of a given SOLAP datacube spatial dimension level are considered in their minimal (i.e. probably smaller than in reality) or maximal extents (i.e. probably larger than in reality), certain types of measures, involving a geospatial parameter related to the members' geometries, are over or under evaluated on that level (e.g. measure "number of kg of pesticide per hectare" which involves the surface of the members' geometries). In that case, the impact is noticeable on the measures values themselves since they vary according to the geospatial parameter. Other measures not involving geospatial parameters may also be poorly evaluated on that level (e.g. number of kg of pesticide per spread zone). In this case, even if the value is computed with the highest accuracy, end-users can still misinterpret that.

For a detailed example, let's say that we record the exact quantity of spread pesticide as being 68kg in a given spread zone. The zone, which is geometrically vague as explained in section 3.3(2), is considered in its maximal extents in the datacube. In such case, we do not know exactly where those 68kg have been spread inside the limits considered. An end-user presented only with the maximal limits and the quantity recorded may think that the quantity has been spread uniformly in the whole extent while it is only in sub-parts of that extent. Some decisions he will make based on such pretty sensitive information may lead him to unexpected results, which means he is exposed to risks of misinterpretation. It is important to qualify that risk to take it into consideration when designing the datacube, especially the spatial data visualization policies. For this example in particular, we can qualify that as a risk of under evaluation of the quantity per area: in fact, the quantity would have been greater for the maximal extents if the spreading where uniform on the whole geometry as unaware users may think.

(2) ***Risk-Aggregation***: It is a risk of misinterpreting the measures' values, related to a given level, risk that is not induced by a potential vagueness on the members' geometries themselves. Instead, it can be induced by:

- *The aggregation formula and/or the geometries involved in the aggregation*: For the pesticide case above (see section 3.3), let us consider the low level Spread Zones (maximal extents) and the level Flood Zones (minimal extents). Only the level Spread Zones members that intersect a Flood Zone member are participating in the aggregations for that member. For example, to compute the total quantity of pesticide to be found a flood zone FZ1, one option would be to "sum" the weighted values related to the results of the "intersection" between FZ1 geometry and level Spread Zones members' geometries (see Figure 3-3). Since the

flood zone is considered in its minimal extents, spread zones that only intersect the maximal extents of flood zones are totally left out of the aggregation. Also, the portion of spread zones that actually intersect the flood zone geometry is the smallest. Since the real limits are somewhere between the minimal extent and the maximal extent, the aggregated values present an uncertainty that is a source for risks of poor evaluation (see Table 3-1). In general, the uncertainty importance depends on the spatial predicates (e.g. intersect, touch, contain) used to select the lower level members that participate in the aggregation for a given higher level member, and the aggregator (e.g. sum, max, average).

- *Existing uncertainty on the measures' values to be aggregated*: In fact, aggregations can simply propagate an already existing uncertainty on the measure's values for a lower level to a given level leading to a risk of misinterpretation on that level.

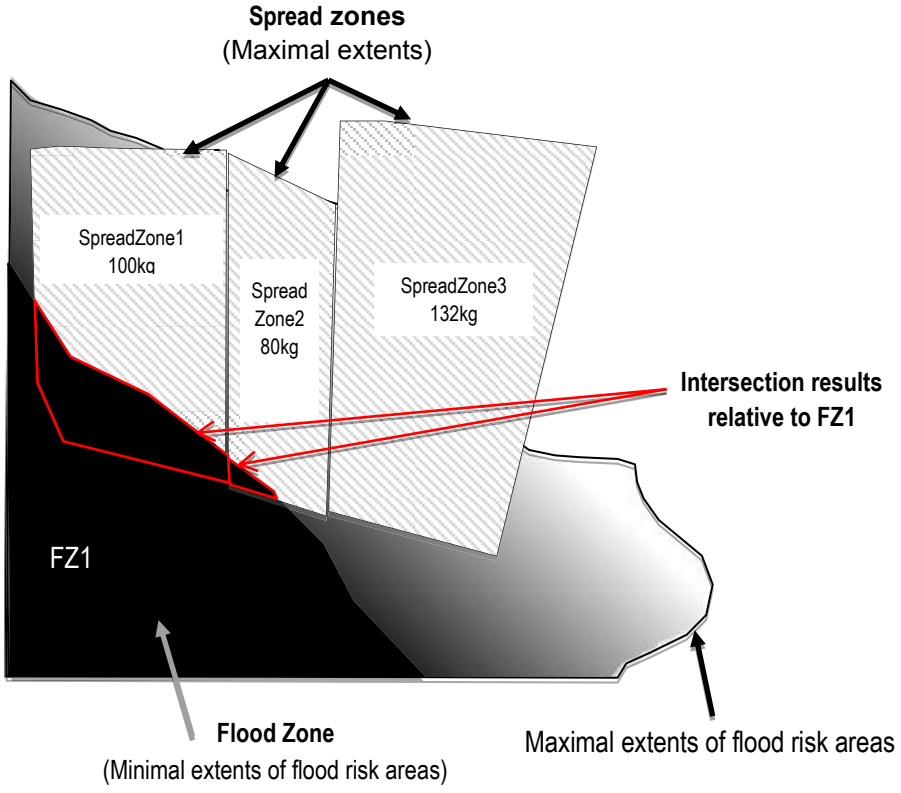


Figure 3-3 : Schema illustrating the aggregation on level Flood Zones

Flood risk areas	Spread zones considered	Aggregation results	Risks of misinterpretation
Minimal extents (**FZ1**)	SpreadZone1 SpreadZone2	Q1 : 23kg Q2 : 20kg	Under evaluation
Maximal extents	SpreadZone1 Spreadzone2 SpreadZone3	Q1 : 240kg Q2 : 98kg	Over evaluation

Table 3-1 : Example of aggregation results for flood zones

We designate by "*Risk-Level"* the risk created for a given level. It is a Risk-Geometry, a Risk-Aggregation, or a combination of both.

3.5 The risk-aware design approach

In this section, we present the requirements of the new risk-aware design approach, followed by the design process advocated by the approach, and then the new concepts and techniques that are needed to accomplish the different steps of the design process.

3.5.1 The risk-aware design approach requirements

The design approach must not only render the resulting SOLAP datacubes Platform Independent Model (PIMs) and physical schemas, but also the aggregation rules, and the visualization policies to use for proper communication regarding the risks of misinterpretation. All those elements must be implementable with existing tools (Spatial Data Base Management System-DBMS, SOLAP server and client). Ideally, the design process should be based on an agile method. In fact, the design process, especially the transformation part, needs to be iterative to allow returns to the key steps of the process at any time during the SOLAP project to refine the design.

3.5.2 The risk-aware design process

It was showed in the literature review (see section 3.2) that to manage the spatial vagueness, different research teams have proposed the use of spatial vague objects models such as fuzzy or exact models to represent the spatial vague data. That means implementing those models in existing SOLAP tools, from the data storage in existing spatial DBMS to the map visualization of the data and results analysis in the SOLAP client, through the analysis of the data, using fuzzy/vague aggregation operators, by the SOLAP server. Even though this approach is the absolute accurate one, it does not always work well in favor of the datacube usability and practical implementation tools are still yet to be available. In this thesis, we aim at dealing with the spatial vagueness issues in a symbiotic trade-off context, and help providing relevant datacubes for any context of application and different end-users. Because the spatial vagueness causes diverse risks of misinterpretation on the datacube (details are to be found in the Spatial vague objects typology), we believe that replacing the spatial vagueness management itself by the management of the

risks of datacube misinterpretation induced by the spatial vagueness is a pertinent way to achieve our goal. The idea here is thus to work out a design process that allows taking the risks into account and delivering datacubes (PIMs) compliant with decision-makers' tolerance to the risks.

To come up with the risk-aware design process, we added new steps to the requirements specification and conceptual design phases of the classic SOLAP datacube design process, exploiting the risk management method advocated by Lévesque (2008) and Gervais, Bédard et al. (2009). By classic SOLAP datacube design process we mean the following steps: Requirements specification (Computation Independent Model-CIM design in MDA method), conceptual design (Platform Independent Model-PIM design in MDA method), and finally the logical and physical design (Platform Specific Model-PSM design in MDA method). The PIMs obtained from the conceptual design phase will be used to design the PSMs in a classic way.

In the rest of this section, we will first make an inventory of the type of actors involved in our risk-aware design process, followed by the description of all the steps advocated by our new process.

3.5.2.1 Actors involved in the risk-aware design process

We consider two main profiles of actors: (Actor profile 1) geospatial systems and data users that are the end-users of the SOLAP application (decision-makers) and systems and data sources end-users such as application domain experts (farmers for example) who actually know, consciously or not, about the data sources quality; (Actor profile 2) the SOLAP experts who have the ability to design and implement SOLAP application.

The people involved in the design process are delegates of users and SOLAP experts in charge of the design. They form a project committee and work together through the majority of the steps of our risk-aware design process. However, the datacubes schemas elaboration and tailoring to tolerance levels are done by the SOLAP experts (profile 2) involved, while the tolerance levels assessment is solely done by the decision-makers (profile 1) involved. In the rest of this section we will designate the users participating in the project committee by *Actor Profile 1* and the SOLAP experts of the committee by *Actor Profile 2*.

3.5.2.2 The risk-aware design process steps

Our Risk-Aware Design Process has two phases: The requirements specification phase (see Figure 3-4) and the conceptual design phase (see Figure 3-5). Each phase includes new steps (in white) that did not exist in the classic design process.

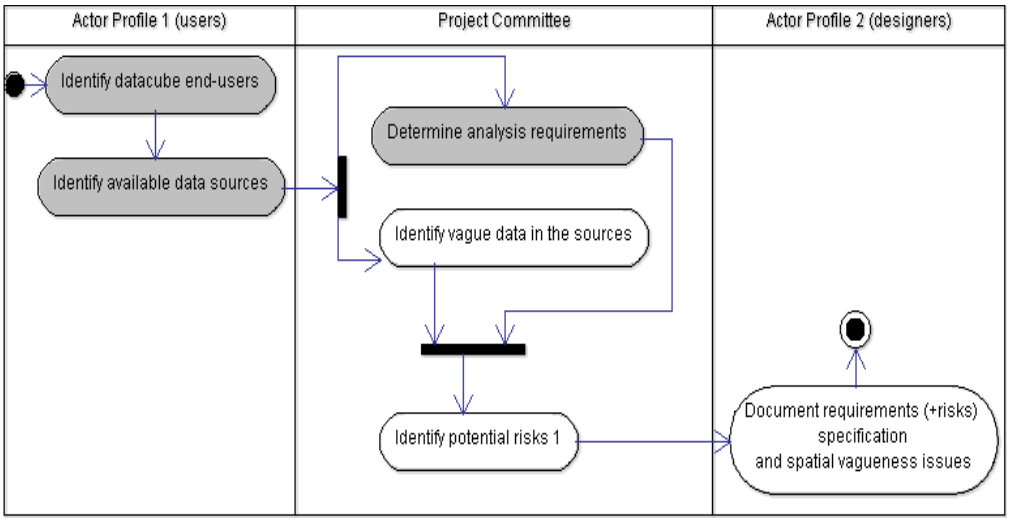


Figure 3-4: Risk-aware design process: Requirements specification phase

Concerning the requirements specification phase, it starts with an end-users' identification (*Identify datacube end-users*), followed by an available data sources identification (*Identify available data sources*), all done by users involved in the project committee. With the results of those two steps, the project committee will work out the analysis needs specification (*Determine analysis requirements*). Then we have the new steps:

- ***Identify vague data in the sources:*** This new step has to be done after the classic data sources identification (*Identify available data sources*). It consists of determining whether sources spatial objects present spatial vagueness issues and monitoring which objects are vague. This will be useful for the tagging of spatial vagueness on the SOLAP datacube multidimensional schema and therefore for intrinsic risks identification later.

- ***Identify potential risks 1***: A first risks identification is done here by the project committee. It is done by exploiting the analysis requirements and the spatial vagueness issues identified ("Risk of wrong interpretation of measures associated to flood zones" for example). It gives the committee a succinct idea of the potential risks of misinterpretation (intrinsic or extrinsic ones). After that, the requirements specification is documented, as well as spatial vagueness issues on the sources and risks identified (*Document requirements (+risks) specification and spatial vagueness issues*), on a CIM for example. That closes the requirement specification phase. One can then move on to the actual conceptual design stage (Figure 3-5), beginning with the elaboration of an initial intended SOLAP datacube PIM (*Develop intended datacube initial PIM + Aggregation rules Visualization policies*).After that, the actual risk-aware design activities start (new steps).

- ***Identify potential risks 2***: In this step, identified risks are updated by the project committee, knowing exactly the elements defined in the multidimensional model (PIM design results) and the

vague spatial data SOLAP datacube end-users will have to deal with in the intended datacube analysis. For example, the “risk of under evaluation of QuantityAsub on level Flood Zones” can be added to the list.

- ***Assess Risks tolerance levels***: Here, the project committee members representing the SOLAP application end-users (decision-makers) are asked to express their tolerance level to each identified risk (e.g. Risk of under evaluation of QuantityAsub on level Flood Zones: Totally Unacceptable i.e. tolerance level = 0). The possible tolerance levels are detailed in the following section 3.6. If all the identified risks are acceptable by them, the current SOLAP datacube PIM is provided (*Deliver intended datacube final PIMs + Aggregation rules Visualization policies*) and it is the end of the design process. If not, we proceed to the following steps.

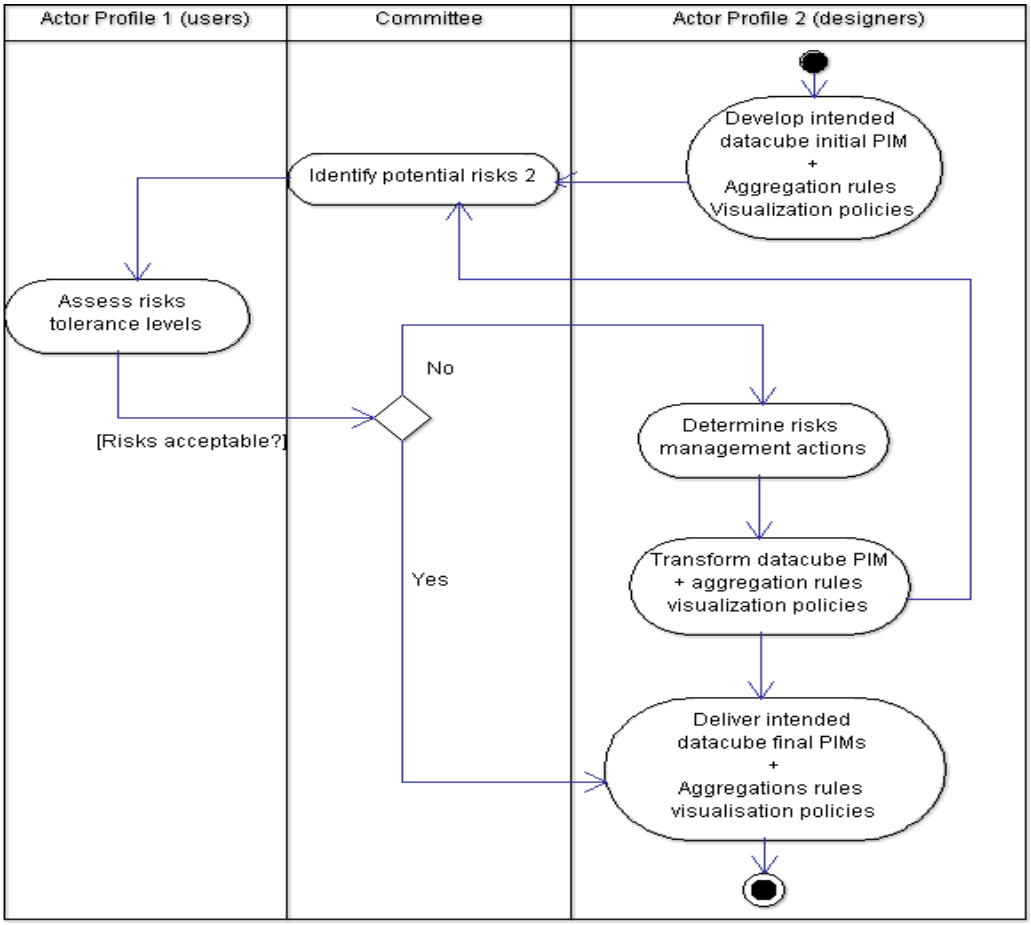


Figure 3-5 : Risk-aware design process: Conceptual design phase

- ***Determine risk management actions***: According to end-users' tolerance levels, different strategies, and thus actions, are defined to reduce the risks of misinterpretation. One action would be for example to remove a level with vague members to avoid the risk of under evaluation of measures values for that level (case of level Spread Zones).

- ***Transform datacube PIM + Aggregation rules Visualization policies:*** Here risks management actions are applied to obtain appropriate SOLAP datacube's PIMs and corresponding aggregations rules and visualization policies. SOLAP experts can decide to only apply some of the identified actions. For example, they can only apply actions that have an impact on the multidimensional structure and/or on other identified risks at first. Then, the process goes back to the Identify potential risks 2 step. If the updated risks regarding the produced datacube PIMs are acceptable for the end-users, the design process is over and the current datacube PIMs are supplied (Deliver intended datacube final PIMs + Aggregation rules Visualization policies); if not, the process goes on again.

The most important question left unanswered now is how end-users can actually assess tolerance levels and how designers can choose risks management strategies and actions. The following section presents the proposals in this regard.

3.6 Risk of misinterpretation assessment and management

Before determining their tolerance to an identified risk, datacube end-users representatives need to evaluate that risk first. In a standard manner, risks evaluation is done by classifying the risks using the Kerzner (2006) risk degrees matrix where the axes are respectively a risk probability of occurrence (High, Medium or Low) and its gravity (High, Medium or Low). According to Kerzner (2006), the risk evaluation step is usually the longest, complex, but also most important one in a risk management process. Here, we have defined a risk degree scale composed of five levels which should be enough to evaluate the risks for a conceptual design purpose. This risk degree scale is a simplification of the Kerzner risk degrees matrix. Those five levels are:

- **Very High Risk**: Probability of occurrence of the risk is High and the severity is High (HH),

- **High Risk**: the risk probability of occurrence is Medium and its severity High (MH) or the probability of occurrence is High and Severity is medium (HM),

- **Medium Risk**: the probability of occurrence is Medium and the severity is Medium (MM),

- **Low Risk**: the probability of occurrence is Low and Severity is Medium (LM) or Probability of occurrence is Medium and Severity is Low (ML),

- **Towards Zero Risk**: Probability of occurrence is Low and Severity is Low also (LL).

Logically, the end-user evaluates the risk level according to his application context. Depending on the risk degree, he can then decide whether he can tolerate a risk or not. He can also decide if he can tolerate that risk directly based on the application context. What matters for our approach is the tolerance level (a qualitative parameter) that the user expresses relative to the risk, no matter at which degree he evaluates the risk beforehand (it's his in-house arrangements).

We define 4 tolerance degrees to fit with 4 risk management strategies as explained a few paragraphs down: 0 for totally unacceptable, 1 for preferably unacceptable, 2 for somewhat acceptable, 3 for totally acceptable (see Table 3-2).

Following risks evaluation, different mechanisms should be executed to manage risks according to the levels of tolerance expressed. According to Gervais, Bédard et al. (2009), there are different ways to cope with identified risks: avoidance, control, transfer or indifference. The *risk avoidance* aims at reducing an unacceptable risk by eliminating the source from which it emerges, the *risk control* aims at reducing the risk by taking preventive actions, the *risk transfer* aims at reducing the risk by transmitting it to a third party such as the end-users or an insurance company and in the *indifference* strategy, the existence of the risk is acknowledged without taking any specific action to reduce it. Moreover, the ISO/IEC-51 (1999) guidelines advocate a risk reduction process that one should follow when choosing a risk reduction strategy. What is noted is that the risk reduction must be assumed by the SOLAP datacube producer first before transferring it to end-users if necessary. That means avoidance and control must be prioritized by the datacube producers. Also, datacube producers must do internal prevention first, then additional protection activities and finally security related information communication.

In our approach, datacube producers are the members of the project committee (the SOLAP experts and delegates of users); delegates of decision-makers in particular are the one deciding if a risk is acceptable or not, using our proposed tolerance level scale.

With all that in mind, we establish a one-to-one relationship between each tolerance level and a risk management strategy by issuing the following hypothesis:

- **Tolerance 0 (totally unacceptable risk): Strategy = Avoidance.** Eliminating a risk source implies doing more or less important internal changes on the SOLAP datacube, so we recommend it for unacceptable risk. Reciprocally, since the risk is totally unacceptable, we find it appropriate to completely avoid the source.

- **Tolerance 1 (preferably unacceptable risk): Strategy = Control.** The control is suggested here not only because it is the appropriate internal prevention strategy to reduce a risk to an

acceptable level, but also because as suggested in the ISO/IEC-51 (1999) guidelines, datacube producers must prioritize control over transfer.

- **Tolerance 2 (somewhat acceptable): Strategy = Transfer.** It is the preferred strategy here because since the risk is somewhat acceptable, we find it appropriate to leave the decision to use the data or not up to the actual end-user. The transfer can be done by properly communicating the risk itself to all end-users so they will make the decision depending upon their particular use case and tolerance.

- **Tolerance 3 (totally acceptable): Strategy = Indifference.** Since the risk is totally acceptable, there is no need to make a particular effort to reduce it.

Even though for each risk reduction strategy, different mechanisms can be applied either before (during data collection, spatial ETL or datacube design) or after the datacube utilization to cope with the risks, in our work we are interested in actions that can take place during the datacube design stage. We distinguish two categories of such actions:

- Actions that change the multidimensional data structure: they are actions that can be taken on the datacube dimensions, hierarchies, aggregation levels, members and choice of measures;

- Actions that do not change the multidimensional data structure: they are actions to apply on the aggregation rules, the visualization policies, etc. without changing the multidimensional structure of the datacube.

Those mechanisms are classified according to the risk management strategies as presented in Table 3-2. The list is not exhaustive, and furthermore, this classification can be used in the future to identify what type of actions one can define to handle the different cases of vagueness presence in datacubes (in measures only, in measures and spatial dimensions etc.)

Risks level (Probability X Severity or vice versa)	Tolerance level	Risk management strategies	Possible actions	
			Impact on multidimensional data structure	**No impact on multidimensional data structure**
Very High Risk (HH)	0: Totally Unacceptable Risk	Avoidance	Delete the risk source: dimensions, levels, members	Delete the risk source: aggregation rules
High Risk (HM)	1: Preferably Unacceptable Risk	Control	Modify members, levels, dimensions;	Add/Modify aggregation rules Prohibit some combinations, define access policies; List the recommended and non-recommended usages in a metadata support; communicate the risk if considered necessary.
Medium Risk (MM)	2: Somewhat Acceptable Risk	Transfer		Communicate the risk
Low Risk (LM)	3: Totally acceptable Risk	Indifference	No action	
Zero Risk (LL)				

Table 3-2: Risk tolerance levels and risks management actions

3.7 Application of the risk-aware design approach to our case study

In this section, we propose to perform the design of the SOLAP datacube of our case study by adopting our risk-aware approach. This is done to illustrate the new approach.

- **Requirements specification phase**

We recall that the datacube end-users are decision-makers in the environmental field (*Identify datacube end-users* step) and the available sources are the pesticide spread activities GIS containing spatial data listed in section 3.3 (*Identify available data sources* step). Also, end-users analysis requirements were presented in the same section (*Determine analysis requirements*).The vague data in the sources are: the area of applied pesticide and flood risk areas (*Identify vague data in the sources* step). The result of the step "*Identify potential risks1*" is shown on Table 3-3. The risks identified at this step are to be characterized after the actual initial datacube is designed. Here they are just collected and expressed in an informal way.

Spatial Object	Definition	Identified risk
Spread zone	Region of applied pesticide (Represented by simple polygons)	End-users may think that every part of the polygons has been spread uniformly with pesticide, which is not the case (spread zones have vague shapes in reality).
Flood risk areas	Region where water can be present in case of inundation. (Represented by simple polygons).	End-users may think that the limits recorded are exact but actually the flood risk areas have broad boundaries.

Table 3-3 : Results of « Identify potential risks 1 »

- **Conceptual design phase**

Develop intended datacube initial PIM + Aggregation rules Visualization policies step: Analysis requirements are expressed on the initial datacube multidimensional model (knowing the available sources) showed on Figure 3-6, and the aggregation rules are shown on *Figure 3-7*. This multidimensional model is defined using the UML profile for SOLAP datacubes modeling presented in Boulil, Bimonte et al. (2011).

This model has: (1) a spatial dimension “Zones” containing regions (represented by simple polygons) on which the quantity of applied pesticides is monitored; the regions are organized according to the hierarchy “Spread Zones → Farms → Flood Zones”; (2) a temporal dimension “Date” with the aggregation levels “Day→Month” and (3) a thematic dimension “Active Substances” (contained in the pesticides) with the unique level “Active substances”.

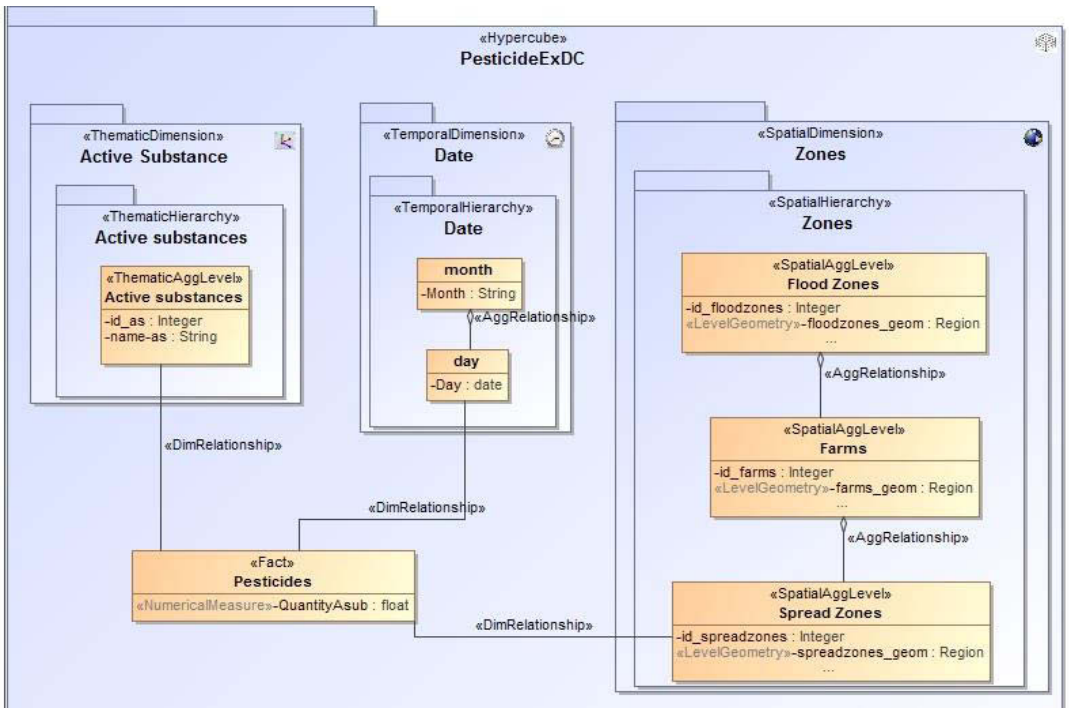


Figure 3-6: Intended datacube multidimensional model

The spatial data to be used in the spatial hierarchies are the spread zones (for level Spread Zones), farms (for level Farms) and flood risk areas (for level Flood Zones). As explained in section 3.3(2), spread zones and flood risk areas have vague shapes in reality. However in our symbiotic trade-off, only simple polygons are used to represent the spatial objects, vague or not. Thus, for the vague spatial objects, the actual geometries that will be stored in the implemented datacube later would be their minimal or maximal extents. In summary, for this initial datacube multidimensional model, we have identified for each spatial level, the corresponding spatial data and the corresponding geometry that is to be stored later in the SOLAP system (see Table 3-4).

Spatial Level	Spatial data	Geometry to be stored in the datacube
Spread Zones	Spread zones as defined in section 3.3(2).	Simple crisp polygons corresponding to the Maximal extents of spread zones as shown on Figure 3-2 (farming plots boundaries)
	They have vague shapes.	
	Ideally represented by vague complex geometries as shown on Figure 3-2	
Farms	Farms as defined in section 3.3(2).	Polygons corresponding to Farms boundaries
	They have well-defined cadastral limits.	
	Represented by simple crisp polygons.	
Flood Zones	Flood risk areas as defined in section 3.3(2).	Simple crisp polygons corresponding to Minimal extents of flood risk areas shown on Figure 3-2 (waterbodies boundaries + buffer)
	They have broad boundaries	
	Ideally represented by vague complex geometries as shown on Figure 3-2	

Table 3-4 : Dimension "Zones" spatial levels description

The expected fact (Pesticides) is described by the measure QuantityAsub that represents the quantity (in Kg) of applied active substances. QuantityAsub will be aggregated along the hierarchies with the aggregation operation Sum to calculate the total quantity of applied pesticide (Top image of Figure *3-7*), or with the aggregation operation Max to calculate the greatest quantity spread (Bottom image of *Figure 3-7*).

Figure 3-7 : Intended Aggregation rules

With this multidimensional model, it is actually possible to answer the SOLAP queries Q1 and Q2 (Cf. section 3.3(1)). Note that Q1 and Q2 expressions become:

- (Q1) *"What is the **TotalPesticide** of **Active Substance** per **Year** for **Flood Zones**?"*

- (Q2) *"What is the **MaxPesticide** of **Active Substance** per **Year** for **Flood Zones**?"*

The visualization policies are at this point the classic ones (plain map and plain cells in pivot table).

Identify potential risks 2: The following Table 3-5 presents a summary of all the identified risks for the pesticide intended datacube.

Levels	Risks	Risks descriptions
Spread Zones (Maximal Extents)	Risk Geometry (*rgSpreadZones*)	Under evaluation
Flood Zones (Minimal Extents)	Risk Aggregation 1 on query Q1 (*raggFZ1*)	Under evaluation
	Risk Aggregation 2 on query Q2 (*raggFZ2*)	Under evaluation

Table 3-5: Risks identified for the Pesticide intended datacube

Assess risks tolerance levels: Using our tolerance level scale, end-users delegates have expressed their tolerance levels to the identified risks as shown in Table 3-6.

Determine risks management actions: According to the tolerance levels, SOLAP experts have chosen appropriate actions among the possible actions presented in Table 3-2. See Table 3-6 for the actions.

Risks	Tolerance level	Actions
rgSpreadZones	0: totally unacceptable risk	Delete Spread Zones level (see Figure 3-8)
raggFZ1	1: preferably unacceptable risk	Modify the members geometries (use the maximal extents)
raggFZ2	1: preferably unacceptable risk	Modify the members geometries (use the maximal extents)

Table 3-6: Risks + Tolerance + Actions for the intended Pesticide datacube

Transform datacube PIM + Aggregation rules Visualization policies: Between the actions chosen in the previous step, SOLAP experts have chosen to apply the one related to rgSpreadZones first, knowing that it will impact the Risk-Aggregations on level Flood Zones. In result, we have the new PIM shown on Figure 3-8 (the Spread Zones level has been deleted).

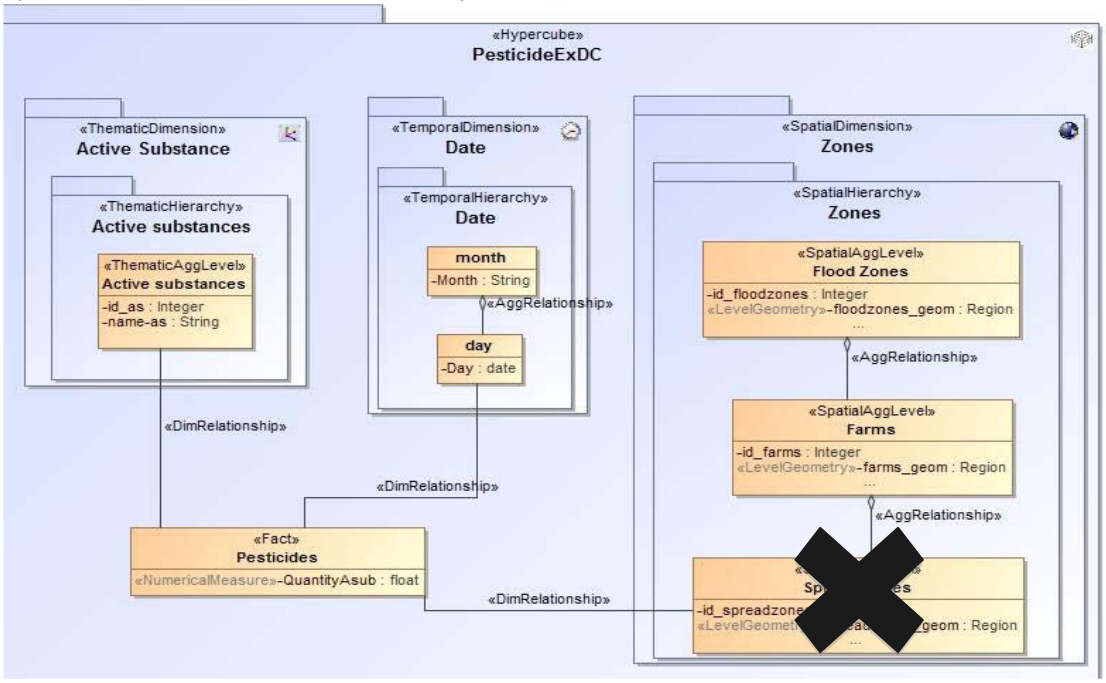


Figure 3-8: Example of final datacube schema – tolerance 0 to Risk-Geometry on Spread Zones (under evaluation)

Return to "Identify potential risks 2" step: The risks are updated (see first column of Table 3-7 and then reassessed (see column New Tolerance Level).

Risks	Old Tolerance level	New Tolerance level	Remarks
~~*rgSpreadZones*~~ *(level deleted)*	0: Totally unacceptable risk		Does not exist anymore (level deleted)
raggFZ1	1:Preferably unacceptable risk	2: Somewhat acceptable risk	It is now a risk of **over evaluation** since the aggregation is supposed to take into account all the farms (new lowest level) that intersect the flood risk areas in their maximal extents.
raggFZ2	1: Preferably unacceptable risk	2: Somewhat acceptable risk	Same as for raggFZ1

Table 3-7 : Risks update and reassessment results

For this datacube, the new actions will simply be risks communication through a visualization policy: color the pivot table cells in red for Flood Zone level. After applying these actions, SOLAP experts go back to the risk identification then the risk assessment steps. The risks are raggFZ1 and raggFZ2 still, and the tolerance levels are still 2 for both.

Since all the risks are acceptable now, the datacube PIM shown on Figure 3-8 is delivered plus the visualization policy defined previously.

3.8 Chapter synthesis

In this chapter, we have proposed a new SOLAP datacube design approach that takes into account SOLAP datacubes risks of misinterpretation and end-users tolerance levels to those risks. First we have defined and classified datacubes risk of misinterpretation. Then, steps of the new design process integrating a risk management method have been detailed. Then we have proposed a risk tolerance levels scale to help the risks assessment by end-users, as well as a classification of the risk reduction strategies that can be adopted to manage the identified risks. More specifically, we have established a one-to-one relationship between each tolerance level and one of the possible strategies to help SOLAP experts narrow the right actions for a given tolerance later. We have presented a small case study to illustrate the interest of our method.

Next, we will address our second objective by working on the elaboration of a risk-aware prototyping method and tool, implementing this approach.

Chapter 4: A Risk-Aware Design Method for Spatial Datacubes Handling Spatial Vague Data: RADSOLAP method

Accepted in International Journal of Data Warehousing and Mining (IJDWM), 2015
Edoh-Alove, E., Bimonte S., Pinet, F.

4.1 Introduction

This chapter addresses the second sub-objective of our research project. Here, we define a new risk-aware design method that implements the principles of our approach: the RADSOLAP method. The chapter main proposals have been submitted and accepted in the International Journal of Data Warehousing and Mining (IJDWM) in April 2015 under the title "An UML profile and SOLAP datacubes multidimensional schemas transformation process for datacubes risk-aware design".

In the previous Chapter 3: we proposed a design process where a risk-management approach was adopted to better deal with the spatial vagueness issues. The tools and principles advocated can be implemented in different ways (in agile methods such as Rapid Application Development, Unified Processing, Extreme Programming etc., in Computer Aided Software Environment-CASE tools, in collaborative frameworks such as the one proposed by our colleague Grira, Bédard et al. (2013) etc.) , however we offer here a rapid prototyping method that aims at helping SOLAP datacube producers throughout the risk-aware design process and validate the design quickly by playing with the prototypes. The method considers the risks of misinterpretation induced by vague data in the SOLAP datacubes and identified by spatial data experts and end-users. It also considers the tolerance levels of end-users to those risks, and based on those parameters, proposes to end-users a set of SOLAP datacubes prototypes iteratively tailored to their tolerance levels. Basically, it allows the multidimensional schemas design by means of UML diagrams, the schemas transformations according to the tolerance levels and actions chosen by SOLAP experts and finally the datacubes prototyping. The proposals detailed in this chapter are: the steps of the method, an UML profile to help conceive the schemas tagged with the spatial vagueness, the risks and the end-users' tolerance levels, a process for the schemas transformations, formal definitions of the schemas transformations functions, and finally the technical architecture of a CASE Tool that can support the method (the SOLAP RADTool).

In Section 4.2of this chapter, we present the state of art related to spatial vagueness management in SOLAP systems and SOLAP datacubes design methods; Section 4.3 briefly describes the ICSOLAP UML

Profile we have extended in Section 5; an agricultural case study is introduced in Section 4.4; Section 4.5 presents our method, including the design process, the RADSOLAP UML profile, the transformation functions and process; finally, Section 4.6 concludes the paper.

4.2 Literature review

The most accurate theoretical approach when it comes to dealing with spatial vagueness issues is probably using vague objects models (fuzzy models, exacts models etc.) to represent the vague spatial data. Some researchers have worked on the practical integration of such models in SOLAP systems. Very recently Jadidi, Mostafavi et al. (2012) have proposed an algorithmic approach based on Fuzzy Set Theory to address fuzzy boundaries of coastal erosion risk areas, among others. Siqueira, Ciferri et al. (2014) have extended the multidimensional model to exploit exact models of vague objects (Bejaoui 2009, Pauly and Schneider 2010). They introduced the "vagueness" in the multidimensional concepts by proposing the VSCube (VS for Vague Spatial) conceptual model. This model supports geometric models associated with membership values and defines the concepts of vague spatial attributes, measures, facts, dimensions and hierarchies, as well as new vague spatial aggregation functions (union, intersection and difference), vague spatial predicates (e.g., intersection range queries or vague spatial range queries), and vague SOLAP operations (vague drill-down, roll-up, slice-and-dice). The VSCube allows the modeling of datacubes exploiting vague spatial data but, no implementation tools have yet been proposed. At this time, there is still much to do to design, implement and exploit SOLAP datacubes exploiting vague objects models. Existing tools (SOLAP server and client) and Spatial DBMS (Pauly and Schneider 2010) are not designed to manage such models and the practical aspect of their integration in those tools is still yet to be developed.

Recently, the Geomatics Community has been interested in preventing spatial data misuse in general. In that vein, a risk of misuse management method has been worked out (Lévesque 2008, Gervais, Bédard et al. 2009), as well as risk management strategies, indicators and frameworks (Gervais, Bédard et al. 2012, Grira, Bédard et al. 2013, Roy 2013), to help users identify and/or assess potential risks of misuses to prevent them during the spatial data usage. In particular, Lévesque (2008) has defined the risk of misuse for the SOLAP datacube in accordance with the ISO (2000) definition of risk, i.e., as being the risk of the probability of occurrence of inappropriate use of a datacube (a usage that leads to unexpected results) combined with the severity of the impact of that inappropriate use. She has also proposed tools to identify and manage the risk of misuse of the intended SOLAP datacube by popping context-sensitive warnings into some multidimensional queries. However, spatial vagueness has not been addressed specifically.

Another study introduced risk management in the database design process (Grira, Bédard et al. 2013). This research focuses on introducing a collaborative approach based on crowdsourcing technology to identify potential risks of data misuse. This approach relies on feedback from end-users about the ways the elements (object class, property, function, association, domains) of a conceptual database design are

defined. Their approach is supported by collaborative tools such as wikis, questionnaires and forums to find the risks identified for the given definitions and to improve these definitions if they decide to. If not, other risk management strategies are adopted. The database design team, not the crowd of end-users, selects the best risk management strategy for each risk identified. This selection may lead to modifications to their database design. Their work is generic for any type of spatial database and, in this regard, can be seen as complementary to the work presented in this thesis.

Jadidi, Mostafavi et al. (2012), Grira, Bédard et al. (2013) and our own research come from the same research group and were thought to be complementary in the way the quality issue is addressed. Other works by Lévesque (2008), Gervais, Bédard et al. (2009) and Roy (2013) also come from our research group and aim at different but complementary objectives.

Regarding the datacube design methodologies different researchers in the OLAP field have put the focus specifically on the extraction of the multidimensional knowledge from either the users' needs (expressed in SQL queries or ontologies – user-driven approach), the available data sources (databases relational or logical models – data-driven approach) or from both users and data sources (hybrid approach). It led to the proposal of multidimensional modeling methods where requirements and conceptual and/or logical datacube schema are derived in an automatic or semi-automatic way according to the type of approach implemented.

The analysis of those design methodologies shows that (spatial) data uncertainty issues are not explicitly considered in the definition of the resulting datacube multidimensional element (i.e. facts, dimensions, hierarchies, measures and aggregations). Instead, uncertainty and datacube quality are principally addressed during the ETL process and/or reporting phase.

4.3 Preliminaries

In this section, we present the main concepts of an UML profile proposed by Boulil, Bimonte et al. (2012) in our research group, which allows the modeling of complex SOLAP applications. This profile is called the ICSOLAP UML Profile. This UML profile will be used to support the SOLAP datacube conceptual design phase in our method as described in Section 4.5.

The purpose of UML profiles is to allow customizing UML for particular domains or platforms by extending its meta-classes (class, property, etc.) (OMG 2011). A profile is defined using three key concepts: stereotypes, tagged values and constraints. A stereotype extends a UML meta-class and is represented using the notation «*stereotype-name*» and/or an icon. For example, it is possible to create a stereotype *«SpatialClass»* that extends the UML meta-class "class". At the model level, this stereotype can be used on classes in UML diagrams to highlight spatial concepts. Tagged values are meta-attributes; that is, the tagged values are defined as properties of stereotypes. Finally, a set of constraints should be attached to

each stereotype to precisely define its application semantics and avoid its arbitrary use by designers in models: for example a constraint can be defined to guarantee that a *«SpatialClass»* class has a geometric attribute called "geom".

The ICSOLAP profile is organized into two main models representing the static and dynamic elements of SOLAP applications: the SDW model and the Aggregation model (Boulil, Bimonte et al. 2012).

The profile description provided in Boulil, Bimonte et al. (2012) mentions that the modelers use the specific stereotypes and tagged values offered by the SDW model to define SOLAP datacube multidimensional elements. In particular, the multidimensional structure is represented as a package stereotyped «*Hypercube»* containing dimensions, also represented as packages. Each dimension is composed of hierarchies, also represented as packages, which organize levels. A level (*«AggLevel»* stereotype) is a class composed of a set of descriptive attributes (*«DescriptiveAttribute»* stereotype) and an identifying attribute. «*SpatialAggLevel»* stereotype designates a level, in a spatial dimension, which has a spatial attribute: this attribute can be geometric (stereotyped «*LevelGeometry»*) or descriptive. A fact is represented using the stereotype *«Fact»*, which is a class with attributes that are measures (*«NumericalMeasure»* stereotype for example).

Usually, aggregations along hierarchies are predefined based on needs in analysis of decision-makers (for example, they want to sum the quantities of sold products by year). That raises the need for conceptual representation of the aggregation operations.

For that purpose, the Aggregation Model offers classes, stereotyped *«BaseIndicator»*, to define a set of aggregation operations to apply to a measure. The measure affected is identified in the «*BaseIndicator»* with the tagged value *«aggregatedAttribute»*, and the aggregation function (*«Aggregator»*) is defined as a parameter of the aggregation operation (stereotyped *«AggRule»*).

4.4 Case study

In this section, we present an agri-environmental case study that will be used in this chapter to illustrate our contributions.

Sewage sludge produced by a wastewater treatment plant can be used as crop fertilizer in agriculture. Farmers spread sewage sludge on cultivated plots to fertilize the soil. Sewage sludge can contain different elements such as trace metals lead, cadmium, chromium, copper, nickel and zinc. Some of those trace metals are essential to the functioning of the biological process (e.g., copper, chromium, nickel). However, at high concentrations, trace metals provided by sludge can become toxic for different forms of life. Thus, from an environmental point of view, it is very important to monitor carefully the activity of sewage sludge spreading.

We consider a SOLAP datacube intended for the analysis and exploration of data related to the sludge-spreading activities in agriculture to help decision-makers in their activities of trace metal concentration control. The SOLAP datacube Platform Independent Model (PIM as defined in the MDA method (OMG 2003)) is shown in Figure 4-1.

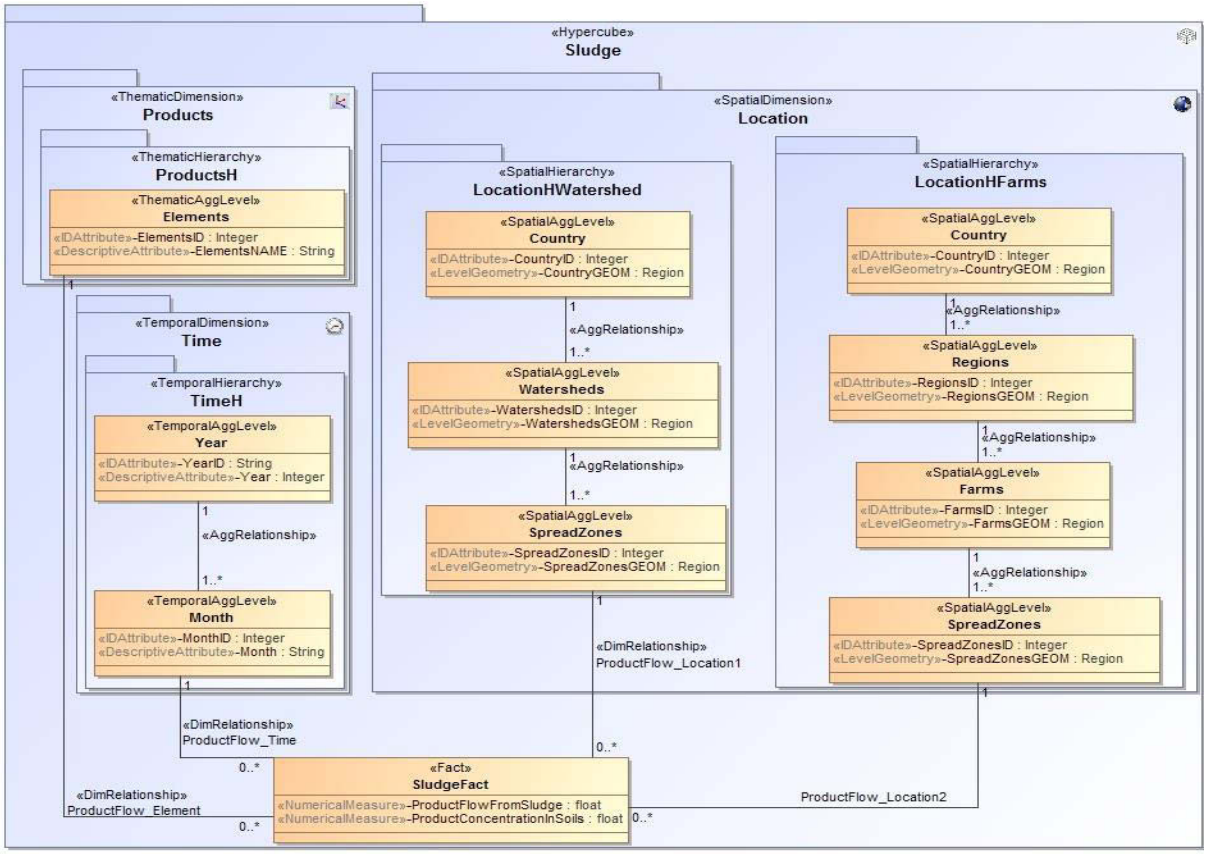


Figure 4-1: Sludge spreading classic SOLAP datacube PIM (modeled using the ICSOLAP UML Profile)

The model presents the following dimensions:

- The dimension *Time* has a hierarchy *TimeH* where months are grouped into years.
- The dimension *Products* has a hierarchy *ProductsH* with the unique level *Elements* (the members being the trace metals: zinc, lead, mercury, etc.).
- In the dimension *Location*, we have: a hierarchy *LocationHWatershed* composed of *SpreadZones*<*Watersheds*<*Country*; a hierarchy *LocationHFarms* composed of *SpreadZones*<*Farms*<*Regions*<*Country*.

The measure *ProductFlowFromSludge* is the flow of trace metals provided by sludge spreading (i.e. quantity of trace metals in grams divided by the area of the developed surface in square meters).

Recorded flow values associated with farming plots are collected from national sludge spreading management GIS and aggregated along the hierarchies using the average function (Figure 4-2).

The measure *ProductConcentrationInSoils* is provided by soils analysis. This measure is the flow of trace metals that is contained in the soils. It is the quantity of trace metals (in milligrams) divided by quantity of soil dry matter (in kilograms) measured at a specific track point taken on farming plots. The values are also collected from national sludge spreading management GIS and aggregated along the hierarchies using the maximum function (Figure 4-2).

Using a SOLAP datacube implementing this model, decision-makers can, for example:

Q1: Aggregate the ProductFlowFromSludge at the watershed level to have an indication of the flow at that level.

This aggregation is described in the BaseIndicator *AVGProductFlowFromSludge* as the *AvgProductFlowInWatershed* aggregation operation. (cf. top image on Figure 4-2):

Q2: Aggregate the ProductConcentrationInSoils at the watershed level to have an indication of the concentration at that level.

This aggregation is described in the BaseIndicator *MAXProductConcentrationInSoils* as the *MaxConcentrationInWatershed* aggregation operation (cf. bottom image on Figure 4-2).

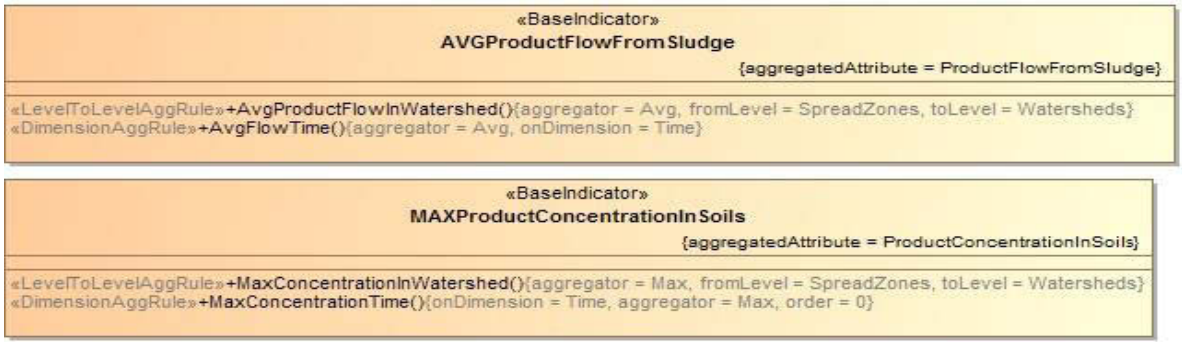


Figure 4-2: Aggregation rules modeling some of the end-user analysis requirements

Spread zones are defined as regions of farming plots where sewage sludge has been spread. In reality, those regions have broad (vague) boundaries.

As shown in the following Figure 4-3, each spread zone can be represented by a geometry composed of:

1. A certain surface (drawn in green), and

2. An uncertain surface which is the space between the certain part and the red boundary.

The green zone is the limit of a farming plot on which sludge has been spread. The geometries of farming plots come from the cadaster system. Because of imprecision related to spreading activity (e.g., the imprecision of the tractor equipment, the impact of winds), it is possible that, in some cases, sludge has been spread outside the green zone. This imprecision is the reason why a larger limit has been defined (the red boundary); we consider that it is not realistic that sludge has been spread outside the red limit during the spreading of the plot.

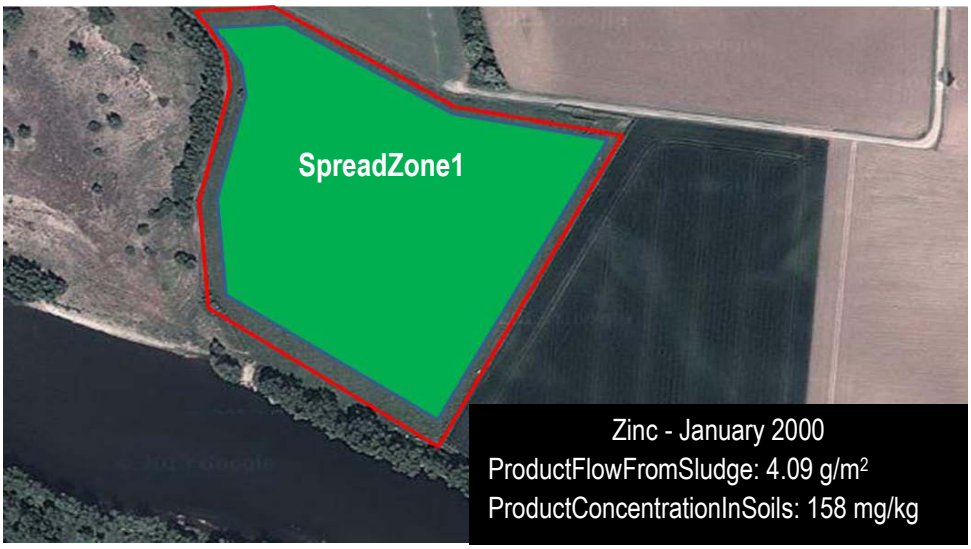


Figure 4-3: Cartographic representation of an example of spread zone spatial vagueness and associated factual data

The spread zone geometries available in the national sludge spreading management GIS correspond to the green zones. Farms are defined as the union of all farming plots of a given farmer. Regions are the administrative regions of the continental France. Watersheds are regions inside of which all surface waters converge to a single point at a lower elevation (rivers, lakes or another water body). Farms, regions and watersheds are well-defined objects with sharp boundaries.

With this intended datacube, when analyzing the *ProductFlowFromSludge* (zinc –January 2000) at the *SpreadZones* level, decision-makers are exposed to the risk of over-evaluating the values. For example, for the *ProductFlowFromSludge* associated with SpreadZone1, decision-makers will think that 4.09 g/m^2 is exact while the real value is somewhere in the interval [3.15, 4.09] (3.15 g/m^2→ red polygon).

We call this risk *Risk-Geometry.*

Definition 1. Risk-Geometry

*A Risk-Geometry is a risk of poorly evaluating a measure on a level that is induced by the vagueness o**n the level geometric attribute**.*

A poor evaluation occurs when a measure is systematically over- or under-evaluated, leading to unexpected analysis and decisions results.

Example 1. Risk of over-evaluation of *ProductFlowFromSludge* on level *SpreadZones*.

When analyzing the same measure at the *Watersheds* level, decision-makers are exposed to the risk of under-evaluating the values. In fact, only the spread zone parts that pour into a given watershed are considered in the aggregation for that watershed. Therefore, the *ProductFlowFromSludge* values are obtained for watersheds by calculating the quantity brought by the spread zone surfaces that are included in the watershed surface. Because the spread zones are vague, there is an uncertainty about the intersections with watershed resulting surfaces, thus an uncertainty on the aggregation results. Knowing that the spread zones are considered to their minimal extent, we can conclude that the intersection surfaces are under-estimated, so the product quantity is under-evaluated as well.

We call this risk, *Risk-Aggregation*.

Definition 2. Risk-Aggregation

A Risk-Aggregation is a risk of poorly evaluating a measure on a level that is associated with the aggregation formula used to compute the measures for that level.

Example 2. Risk of under-evaluation of measure *ProductFlowFromSludge* on *Watersheds* level.

There are other risks of poor evaluation associated to the intended SOLAP datacube (the complete list can be found in List of risks of misinterpretation associated with the Sludge SOLAP datacube). Reducing those risks early on during the datacube design will provide decision-makers with appropriate SOLAP datacubes where analysis errors are minimized. Motivated by this need, we propose a new risk-aware SOLAP datacube rapid prototyping method in the following section.

4.5 Rapid Prototyping of SOLAP datacubes: RADSOLAP method

The main idea of our method is to explicitly manage the risks of measures poor evaluation during the design process. In the majority of cases, those risks of poor evaluation cannot be avoided when exploiting spatio-multidimensional data (measures and spatial level members) that are marked by spatial vagueness in classic SOLAP systems (no use of complex vague object models). The method implements the general risk-aware design approach we have advocated in Edoh-Alove, Bimonte et al. (2015). The management of the risks of misinterpretation is done in three main steps: first, the risk is identified, then a risk management strategy (avoidance, control, transfer or indifference as defined in Gervais, Bédard et al. (2009)) is chosen according to the end-users tolerance level (Totally unacceptable (level=0), somewhat

unacceptable (level=1), somewhat acceptable (level=2) and totally acceptable risk (level=3) to the risk and finally a risk reduction action is applied to the multidimensional schema. The actions can modify the multidimensional structure (e.g. level deleting), the aggregation formulas (e.g. aggregator modification) or can consist of the definition of visualization policies variables (e.g. pivot table cells colors) to communicate the risk to the end-users. Further explanations on the risks management approach can be found in Edoh-Alove, Bimonte et al. (2015).

The proposed method should:

I. Use a data model representing vague spatial data with simple geometries (point, line, polygon) to allow a feasible implementation in existing SOLAP systems;

II. Explicitly support SOLAP datacube aggregations and the definition of visualization policies variables, as well as spatio-multidimensional schemas definition. Indeed, our method should provide not only the SOLAP datacube multidimensional and physical schemas as outputs but it should also specifies the different pertinent and authorized aggregation operations, as well as visualization policies variables values (e.g. red color for *SpreadZones* level cells) to communicate the risks if needed;

III. Allow the possibility of an implementation according to the rapid prototyping paradigm (Bimonte, Nazih et al. 2013): the method should facilitate the roll-back to some of the key steps of the design to revise the choices made and refine the datacube modeling. Because our design method will define visualization policies variables values and change the spatio-multidimensional model, the SOLAP application end-users need to “play” with prototypes to validate the resulting datacubes, before SOLAP experts move to the real implementation.

The steps of our RADSOLAP method are (see Figure 4-4):

1. Users informally define the SOLAP functional requirements and semantics, identifying vagueness in spatial data sources.

2. SOLAP experts create a datacube Platform Independent Model starting from the analysis needs of the users defined in step 1. Vague geometric members and measures are then identified on this initial model. For example, in our case study, spread zones are identified as being vague.

3. Users informally identify and assess risks associated with the multidimensional elements of the model marked by spatial vagueness. In our case study, for example, the Risk-Geometry of over-evaluating the product flow brought on spread zones is identified, and a tolerance level of 0 (totally unacceptable risk) is expressed.

4. SOLAP experts extend the previously defined spatio-multidimensional schema by adding information on risks and associated tolerance levels to the model.

5. SOLAP experts choose a set of risk management actions (e.g. delete a level, modify aggregator, communicate the risk visually for the end-user), according to the tolerance levels defined by end-users, between possible actions associated with a given tolerance levels. Those actions are then applied on the PIM to create a new version in an iterative process that calls for risk reassessment by the users. SOLAP experts can choose first to apply the actions that modify the multidimensional structure or that have an impact on other identified risks. Afterwards, the residual risks will be reassessed by the users and then managed in a second iteration, and so on. Ultimately, the resulting PIMs are automatically translated into physical schemas and prototyped. For example, for the Risk-Geometry on spread zones, and a tolerance level of 0 (totally unacceptable risk), SOLAP experts can choose the avoidance strategy, thus the action "delete level" in a list of possible actions corresponding to that strategy; this action will delete the whole *SpreadZones* level in the spatial dimension of the spatio-multidimensional datacube model (Figure 4-1)

6. The prototype is fed manually with realistic sample data.

7. Users, more specifically decision-makers, access and explore these sample data using simple pivot tables of the SOLAP client, to validate the prototype.

    - If the prototype is accepted by end-users but visualization policies variables values choices for risk display are not, then return to step 5 to test other actions.

    - If the prototype is not accepted by users, return to step 1 if tolerance levels cannot be changed, otherwise return to step 3.

For example in our case study if the prototype is not accepted, return to step 1 and choose to consider the spread zones in their maximal extent.

8. Once the prototype is accepted by users, data are collected, ETL is designed, and the prototype is engineered by the SOLAP experts.

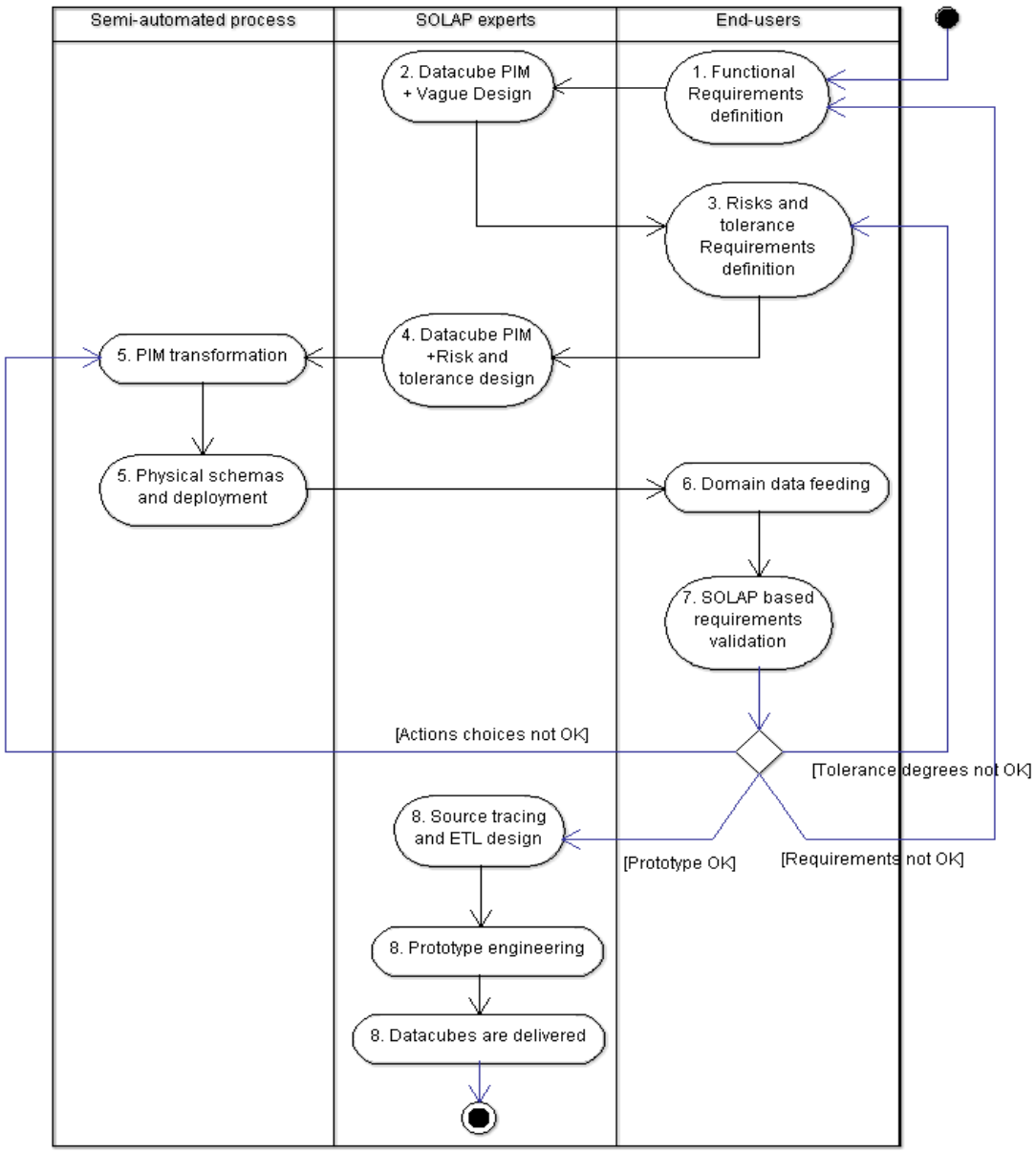


Figure 4-4: RADSOLAP method

Next, we present in Section 4.5.1 the UML profile that supports the PIM design phase (steps 1 through 4 of the method), and in Section 4.5.2, we present the formalization of datacube PIMs transformation functions that support step 5 of the method.

4.5.1 RADSOLAP UML Profile

It has been recognized that conceptual models are mandatory during the design phases of the datacube (Rafanelli 2003) and spatial database (Parent, Spaccapietra et al. 2006) development process, and designers often use UML to describe those models. Thus, in our approach we want to provide SOLAP

experts with a visual formal language to model SOLAP datacubes enriched with information about the spatial vagueness, the risks of poor measure evaluation and the tolerance level.

We call the datacube schema enhanced with such information the *RiskHypercube*.

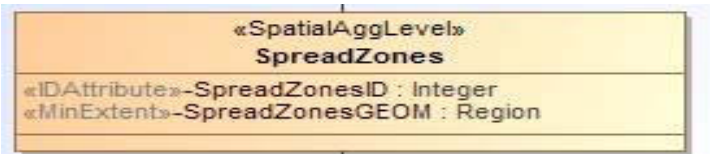
Definition 3. RiskHypercube

A RiskHypercube Cr is a datacube schema that has at least one spatial level SL or measure M with vague geometries AND metadata on the risks and tolerance levels associated with that spatial level or measure.

For that purpose, we have extended the ICSOLAP UML profile presented in Section 4.3. The complete UML Profile meta-model can be found in the RADSOLAP UML Profile metamodel. We defined a new stereotype for the hypercube called «*RiskHypercube»*. A *RiskHypercube* is a specialization of *Hypercube*. Additionally, to support the vagueness expression on the initial PIM as advocated in step 2, we defined different stereotypes to use for level geometric attributes or for geometric measures. Those stereotypes were defined according to the method recommendations (section 4.5) and based on the vague data representation approach introduced in Bejaoui (2009):
«*MinExtent*» when the minimal extent of the phenomenon is considered: it is, for example, the case of the geometric attribute «*SpreadZonesGEOM»* (green regions on Figure 4-3), in the *SpreadZones* class, as shown in Figure 4-5;
«*MaxExtent*» and «*ExactExtent*» for maximal (red polygons in Figure 4-3) and exact extents (for geometries that are spatial vagueness-free such as watersheds, farms and region geometries in our case study), respectively.

Figure 4-5: Level SpreadZones with one of our new geometry stereotypes («MinExtent»)

For the risks and tolerance levels expression on the datacube PIM (step 4), a class stereotyped «*RiskLevel»* is introduced to allow the association of the identified risks with the appropriate level on the PIM (see Figure 4-6). The *«RiskLevel» class* contains attributes stereotyped *«RiskGeom»* or «*RiskAgg»* that handle the identified risks. For example, we associate with the *Watersheds level,* the class *RiskWatersheds, which* contains the attributes *raggSludgeWatershed* and *raggSoilsWatershed that* hold



the *Risk-Aggregations* on *Watersheds.*

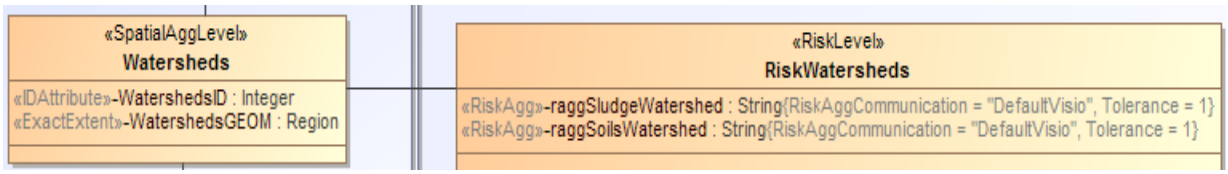


Figure 4-6: RiskLevel on the Watersheds level

To allow the adding of the tolerance level of the end-users to each identified risk, the *«RiskLevel»* attributes are tagged with Tolerance values (Tolerance = t). For example in the *RiskWatersheds class* (*«RiskLevel»* class associated with *Watersheds*), we have the attributes: *raggSludgeWatershed* with the tolerance level of 1 (see Figure 4-6).

As stated previously, our design method has to define risk communication policies (requirement (II)). To allow this definition, we added two types of tags to *«RiskLevel»* attributes to hold each risk communication policy. It is the *«RiskGeomCommunication»* and *«RiskAggCommunication»* (intended for the map visual variables, pivot table or charts styling) tags. For example, in the *RiskWatersheds* class, we have the attribute *raggSludgeWatershed* with a *«RiskAggCommunication»* tag. The values of those tags are at "DefaultVisio" which corresponds to the basic map visualization with no risk to communicate. The attributes will be changed later during the schema transformation process.

Because *Risk-Aggregations* depend on the aggregation operations, the aggregation rules have a new tag («create») to express which risk they create. That way, a relationship is created between the rule and the *Risk-Aggregation*, and if the rule is removed, the risk also disappears. For example, in aggregation rule *AVGProductFlowInWatershed* «create» points to *raggSludgeWatershed* (see Figure 4-7).

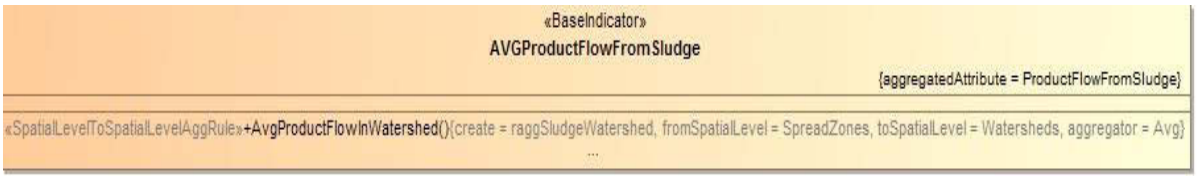


Figure 4-7: Risk-Aggregation tagging in base indicator AVGProductFlowFromSludge

Example 3.

The *RiskHypercubeSludgeRisk* schema presented on the following Figure is a datacube schema where we have a level with vague geometries (*SpreadZones* level) and information about the risks and associated tolerance levels described in *«RiskLevel»* classes *RiskSpreadZones*, *RiskWatersheds* and *RiskFarms*.

4.5.2 Datacube PIM transformation

In the step 5 of our method, SOLAP experts need to conduct SOLAP datacube PIM transformations. For a given risk, and for a given tolerance level expressed on that risk, they have to choose the appropriate action (delete or modify a datacube multidimensional schema element, modify visualization policies variables values to add quality indicator for a measure or to communicate a risk regarding the aggregation levels etc.) to reduce the risk and design adapted datacubes.

The datacube models are to be transformed on the fly and in an iterative manner. So after performing actions that have modified the datacube, it is important to ensure that the models are still valid from a multidimensional point of view. Only valid datacubes are delivered at the end. We consider that to be **valid**, the resulting **datacube models must respect the spatio-multidimensional model logical consistency, allow aggregations and fit user needs in analysis**.

Trying to guarantee the datacube model validity, one will face different issues:

1. How to make sure that with the changes applied to manage each risk, the datacube models still fits user needs in analysis?

   In our case study, we have two different measures: *ProductFlowFromSludge* and *ProductConcentrationInSoils*. If decision-makers have a tolerance level of 0 for the Risk-Aggregation related to *ProductFlowFromSludge* on *SpreadZones*, an action of deleting the level *SpreadZones* will deprive the users of the analysis (all *BaseIndicators*) on the *ProductConcentrationInSoils*.

2. How to make sure the datacube multidimensional structure still allows aggregation each time one action is applied to manage the risk on a given level?

   In our case study, we have an alternative hierarchy in the spatial dimension. If the lowest level (SpreadZones) is deleted during the transformations, one will need to change the hierarchy by creating another lowest level; otherwise the model will not respect the multidimensional logic (see Figure 4-8), and aggregation will not be possible with the new multidimensional structure.

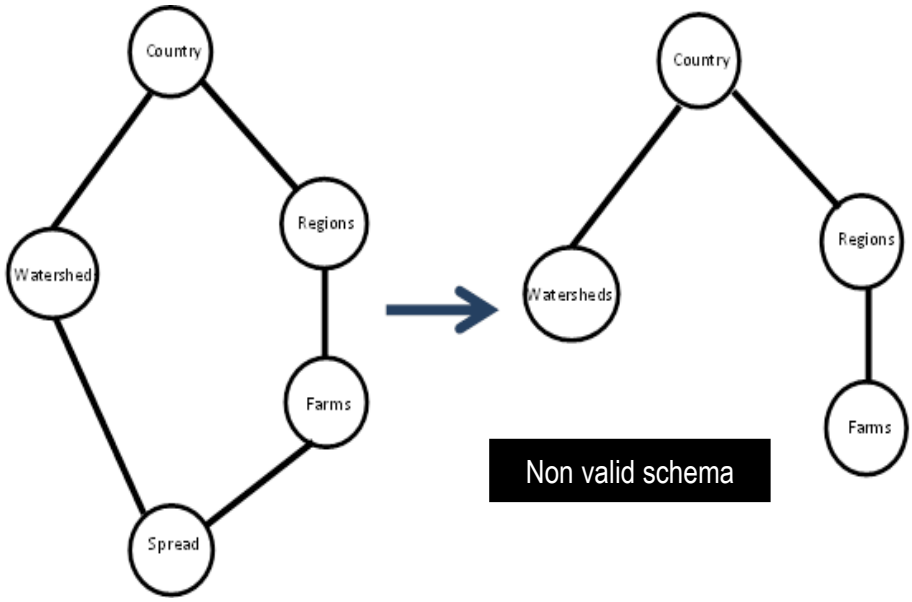


Figure 4-8: Illustration of transformation-related issue 3

One way to resolve those issues is to satisfy the two following requirements: *indicator-level dependency* and *well-formed hierarchy*.

Definition 4. Indicator-level dependency

An indicator-level dependency is respected in a datacube model when each spatial level is associated with only one pair measure/BaseIndicator in the datacube model (see Figure 4-9).

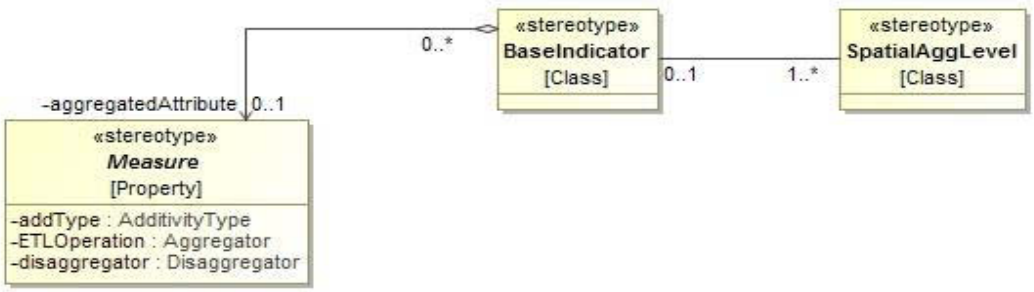


Figure 4-9: Indicator-level dependency

Definition 5. Well-formed hierarchy

To be well-formed, a hierarchy must be represented as a partial order with a bottom level (Pedersen and Tryfona 2001).

Example 5.

The spatial hierarchy shown on the left image of Figure 4-8 is a well-formed hierarchy because it is represented as a partial order with the bottom level *SpreadZones.*

Now, we define the concept of *elementary SOLAP datacube*, on which is based the transformation process and functions in our method.

Definition 6. Elementary SOLAP datacube

An elementary SOLAP datacube is a datacube where:

- each spatial hierarchy is a total order,
- there is a unique measure,
- there is a unique BaseIndicator.

Such a datacube respects the indicator-level and the well-formed hierarchy requirements:

- Well-formed hierarchy: Because a total order is also a partial order, the elementary SOLAP datacube spatial hierarchies are well-formed by definition.
- Indicator-level dependency: By definition, an elementary datacube holds a unique pair measure/BaseIndicator. Thus each spatial hierarchy is associated with that unique pair, meaning that the indicator-level dependency is respected.

Example 6.

The datacube shown in Figure 4-10, below, is an elementary datacube that has a spatial hierarchy *LocationHWatershed* in a total order (*SpreadZones*<*Watersheds*<*Country*), a unique measure *ProductFlowFromSludge* and a unique *BaseIndicator AVGProductFlowFromSludge* associated with the measure.

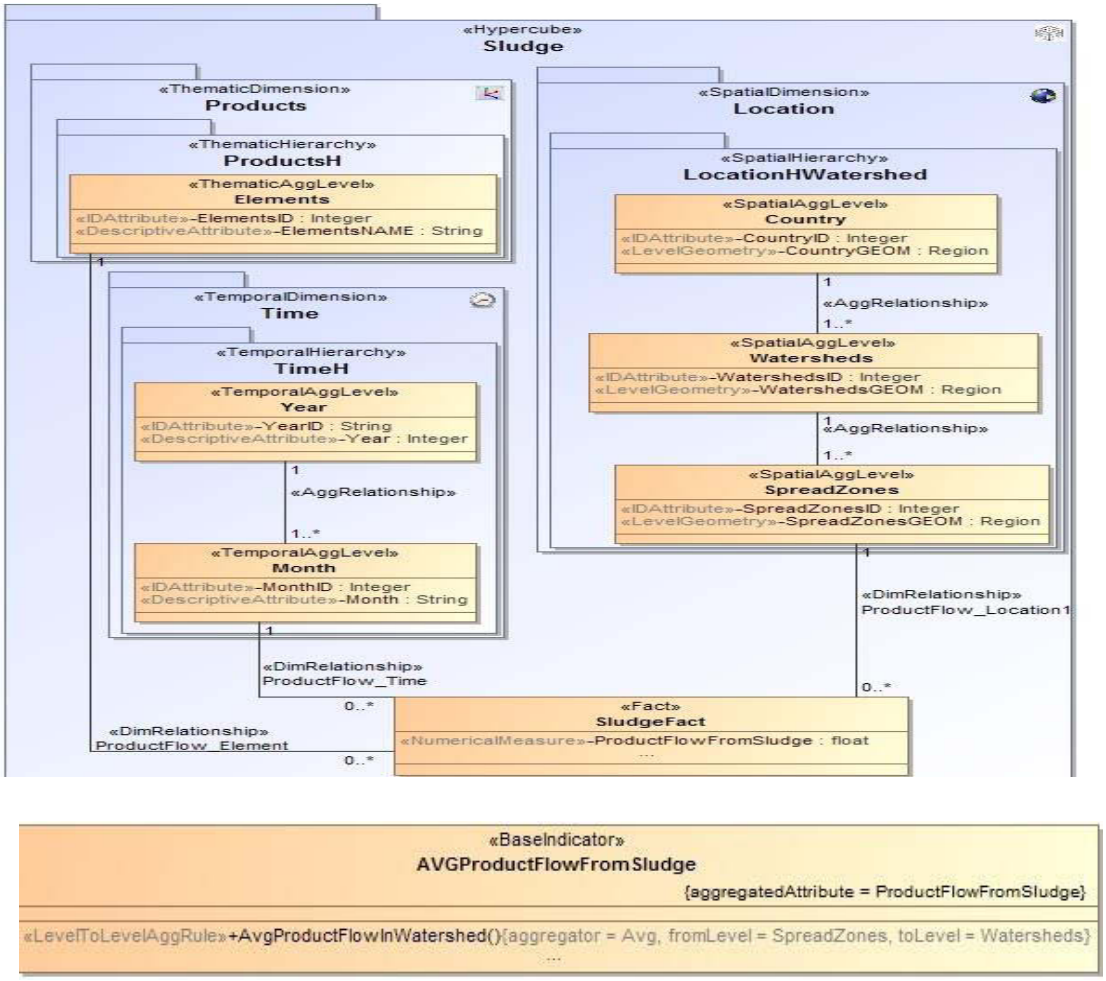


Figure 4-10: Example of an elementary SOLAP datacube

An elementary datacube model is a valid datacube as long as it fits the need in analysis of the end-users.

In the next section, we present the formal definition of transformation functions that help apply the risk management actions to elementary datacube models.

4.5.2.1 Datacube schema transformation functions

We have formalized the basic datacube schema transformation functions that are necessary to technically apply the actions that impact the multidimensional structure, the aggregation rules and the risk communication parameters. The definition of the functions is supported by the RADSOLAP UML profile presented in the previous section.

Each function takes as the main input a *RiskHypercube* modeling an elementary datacube and delivers as output another *RiskHypercube*. The following table (Table 4-1) lists the different functions identified and their descriptions. This list does not pretend to be exhaustive in this thesis.

There are two classes of functions:

- *Functions that change the datacube multidimensional structure*: they allow the modification or removal of multidimensional elements such as dimensions, levels, level attributes (e.g, *DeleteSpatialDimension*, *ModifySpatialLevel*). The addition of levels, dimensions or aggregation rules implies the changing of analysis requirements. For that reason, we do not recommend the use of such actions at this step to manage the risks. Requirement changes should be made at the *Functional requirements analysis* step.

- *Functions that do not change the datacube multidimensional structure*: they aim at modifying datacube model elements such as aggregation rules, risks and risk communication attributes (e.g, *ModifyRiskAgg*, *ModifyAggRuleAggregator*).

Functions	Description
Functions that change the datacube multidimensional structure	
DeleteSpatialDimension	Delete a whole spatial dimension
DeleteSpatialLevel	Delete a level which is neither the leaf nor the root of the hierarchy
DeleteLowestSpatialLevel	Delete the lowest level (root)
DeleteHighestSpatialLevel	Delete the highest level (leaf)
ModifySpatialLevel	Modify the attributes of a spatial Level
Functions that do not change the datacube multidimensional structure	
ModifyAggRuleAggregator	Modify the aggregator defined in an aggregation rule
ModifyAggRuleFromLevel	Modification the spatial level from which the aggregation is done to obtain the values for a given spatial level
ModifyRiskAgg	Modify risk-aggregation attribute
ModifyRiskGeom	Modify risk-geometry attribute
ModifyRiskGeomComm	Modify risk-geometry communication attribute
ModifyRiskAggComm	Modify risk-aggregation communication attribute

Table 4-1 : Datacube schema transformation functions

In the following paragraphs, we present the formalization of the *DeleteLowestSpatialLevel* function.

We consider Cr, a RiskHypercube with one dimension Ds, holding one spatial hierarchy Hs, one measure M and one associated BaseIndicator B.

Hs = SpatialLevel [0] < SpatialLevel [1] <… <SpatialLevel [n], with SpatialLevel [i] (i $\in$ {0, 1…n}) being the aggregation levels at the position i in the hierarchy Hs.

Hs has n-1 aggregation relationships *AggRel[i], (i ∈ {0, 1…n}, n* being the number of spatial levels. AggRel[i] is a relationship between SpatialLevel[i] and SpatialLevel [i+1].

B = {AggRuleSL[i], i ∈ {1… n}} such as AggRuleSL[i] is the aggregation formula defined for a spatial level at the position i in the hierarchy Hs, and n is the number of spatial levels.

A SpatialLevel[i] can have an associated class RiskLevel that describes risks of misinterpretation in Cr.

For our illustrations, we will take this following Cr example (Cf. Figure 4-11):

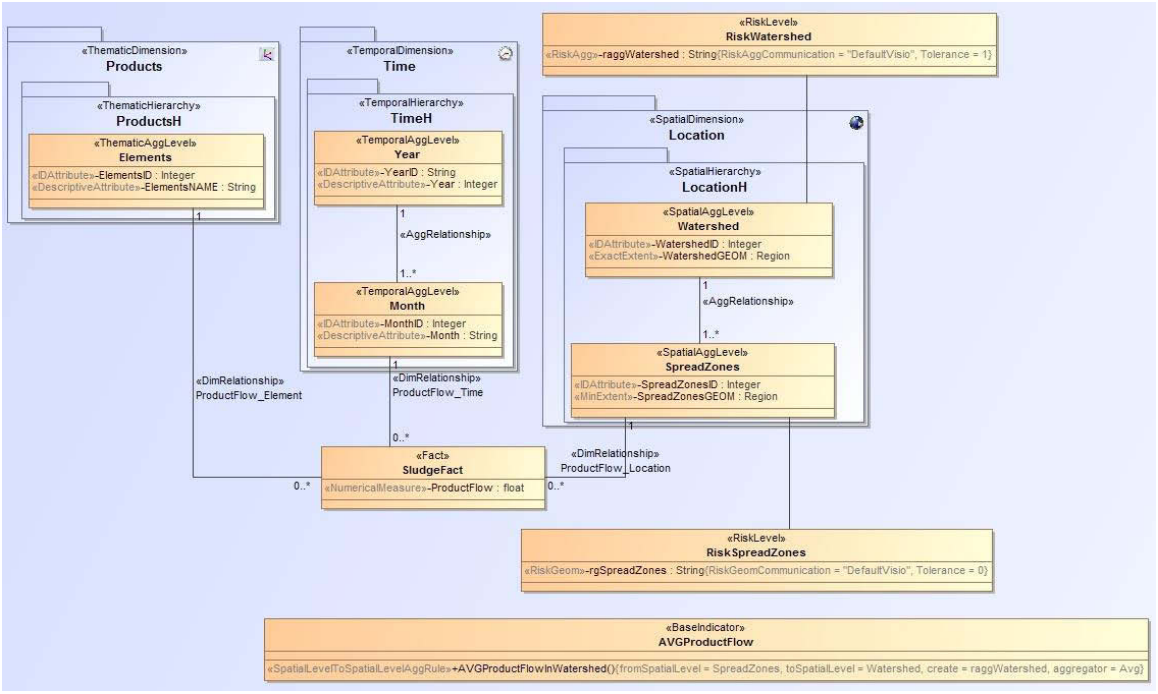


Figure 4-11: Example of elementary RiskHypercube: SludgeCr

Definition 7. DeleteLowestSpatialLevel Function

This function deletes the lowest level of the hierarchy. It takes as input a RiskHypercube and the dimension from which the lowest level should be deleted.

The output is a new RiskHypercube where the level, the associated RiskLevel, the related aggregation relationship and aggregation rules in the BaseIndicator are all deleted. The RiskAgg attributes that are indexed by the tag “create” in the deleted aggregation rules are thus tagged with the tolerance value of 99, meaning that the risk information is invalidated for that level.

DeleteLowestLevel (RiskHypercube *Cr,* **SpatialDimension** *Cr.Ds)* **= RiskHypercube** *Cr’*

Such as,
Cr' = Cr – {Cr.Ds.Hs.SpatialLevel[0], Cr.Ds.Hs.AggRel[0], Cr.Ds.Hs.SpatialLevel[0].RiskLevel, Cr.Ds.Hs.DimRel}
– Cr.BaseIndicator.AggRuleSL[i], i ∈{0,k}, k being the number of aggregation rules to be deleted, where Cr.BaseIndicator.AggRuleSL[i].fromSpatialLevel is SpatialLevel[0]

with new Cr.Ds.Hs.DimRel such as Cr.Ds.Hs.DimRel.identifiedFact is Cr.F and Cr.Ds.Hs.DimRel.identifyingLevel is Cr.Ds.Hs.SpatialLevel[1].
and Cr.Ds.Hs.[Cr.BaseIndicator.AggRuleSL[i].toSpatialLevel.value].RiskLevel.Ragg.tolerance = 99 for each i ∈ (Boutry, Gassion et al.) such as
Cr.Ds.Hs.[Cr.BaseIndicator.AggRuleSL[i].toSpatialLevel.value].RiskLevel.Ragg.name = Cr.BaseIndicator.AggRuleSL[i].create.value

Example 7. Deleting the lowest level of SludgeCr (level SpreadZones). The result of this operation is shown in Figure 4-12 below.

DeleteLowestLevel (SludgeCr, SludgeCr.Location) = SludgeCr'
Such as
SludgeCr' = SludgeCr – {SludgeCr.Location.LocationH.SpreadZones, SludgeCr.Location.LocationH.ProductFlow_Location, SludgeCr. AVGProductFlow.AVGProductFlowInWatersheds}
With new SludgeCr.Location.LocationH.ProductFlow_Location and SludgeCr.Location.LocationH.[SludgeCr.AVGProductFlow.AVGProductFlowInWatersheds.

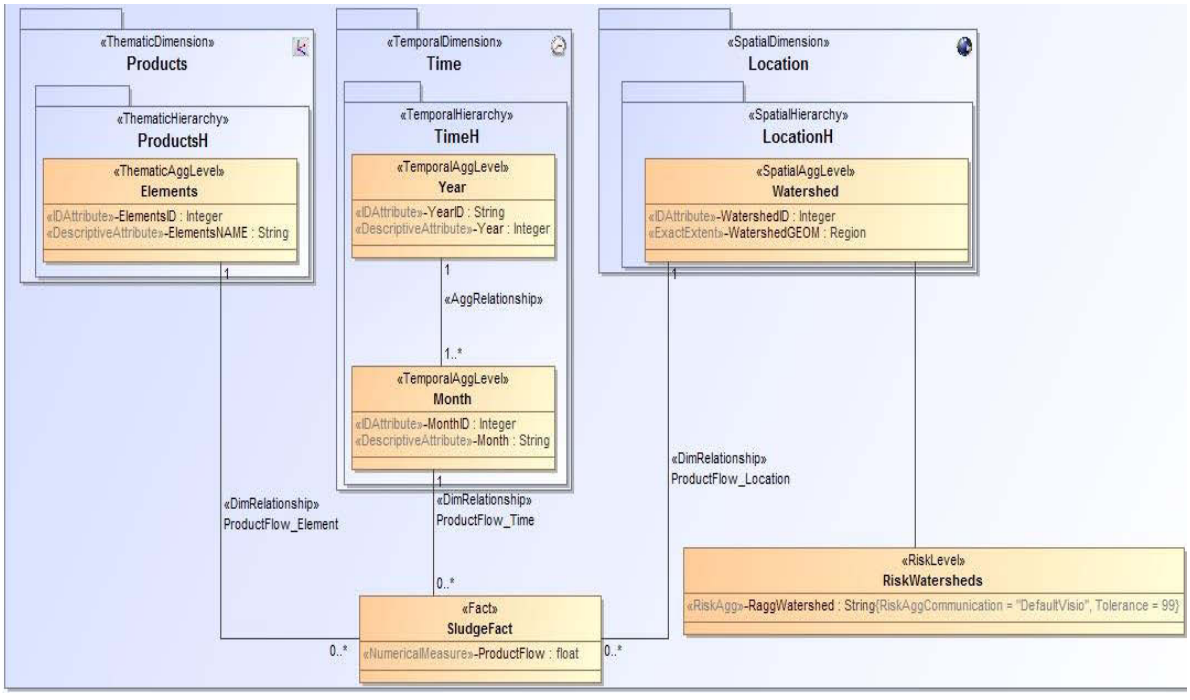


Figure 4-12: Result of the application of the DeleteLowestLevel function on SludgeCr

The remaining defined functions can be found in SOLAP datacube PIM transformation functions.

An elementary SOLAP datacube remains elementary after transformations.

An elementary SOLAP datacube is defined as a datacube where:

(1) *each spatial hierarchy is a total order,*

(2) *there is a unique measure,*

(3) *there is a unique BaseIndicator.*

The transformation functions defined delete or modify spatial levels, aggregation rules, risk attributes and risk communication attributes. Applied on a valid elementary datacube, we will have as output a datacube that still has the following features:

(1) Each spatial hierarchy is a total order: the function that impacts the spatial hierarchy is the delete of spatial level. Whether it is the lowest, highest or any other level that is deleted, the aggregation relationships are corrected in consequence so the hierarchy remains a total order with one bottom level.

(2) A unique measure: no function deletes or adds measures, so the resulting datacube model still describes only one measure.

(3) A unique BaseIndicator: No function adds BaseIndicators and even though aggregation rules are deleted, the BaseIndicator still hold aggregation rules related to the new hierarchy. Therefore, the resulting datacube model still presents a unique BaseIndicator. If there is only one spatial level left in the hierarchy and no BaseIndicator, it is a particular case of elementary datacube because there is still one unique measure.

With that, we have proven that an elementary datacube remains elementary through any transformation or combination of transformations.

4.5.2.2 Datacube schema transformation process

To simplify the comprehension of the proposed transformation process, we consider only one identified risk for our illustrations: Risk-Geometry of over-evaluating m1 on *SpreadZones* with the associated tolerance 0 and the action chosen “Delete level”. The overall transformation process built upon the elementary datacube hypothesis is the following (cf. Figure 4-13):

First, the process takes the initial datacube PIM and splits it into N elementary datacubes to guarantee the datacube model validity throughout the transformations (step 1). After that, each elementary datacube is transformed by means of the transformation functions described in the previous section, according to risk

management actions chosen (step 2). Finally, to reduce the number of final adapted datacubes to deliver to the prototyping steps, a fusion method is applied on the transformed datacubes.

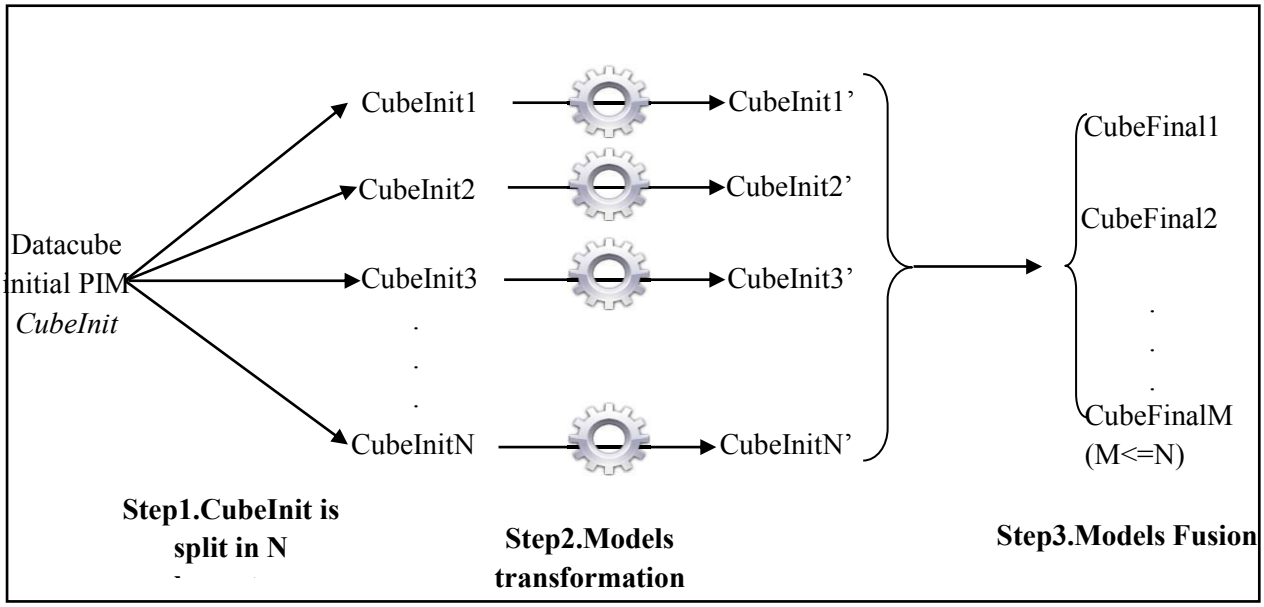


Figure 4-13: SOLAP datacube model transformation process

Step1.*In this step, the initial model is split into N elementary datacube models (CubeInit1…, CubeInitN).*

Definition 8. Datacube Splitting

The datacube splitting is a process where an initial datacube model with one spatial dimension, H spatial hierarchies in that spatial dimension, M measures, Bj (j from 1 to M) BaseIndicator for each measure creates N = H x $\sum_{j=1}^{M}(Bj)$ *elementary datacube models CubeIniti.*

Each CubeIniti, i from 1 to N, represents some of the analysis needs of the users. Aggregation rules with associated risks on spatial levels present in each CubeIniti are kept for that datacube.

Example 8.

In our case study, we have H = 2 spatial hierarchies, M = 2 measures and B1 = 1 BaseIndicator for measure m1 and B2 = 1 for measure m2. The split process will return N = 2 x (1+1) = 4 new datacube models as shown in Figure 4-14.

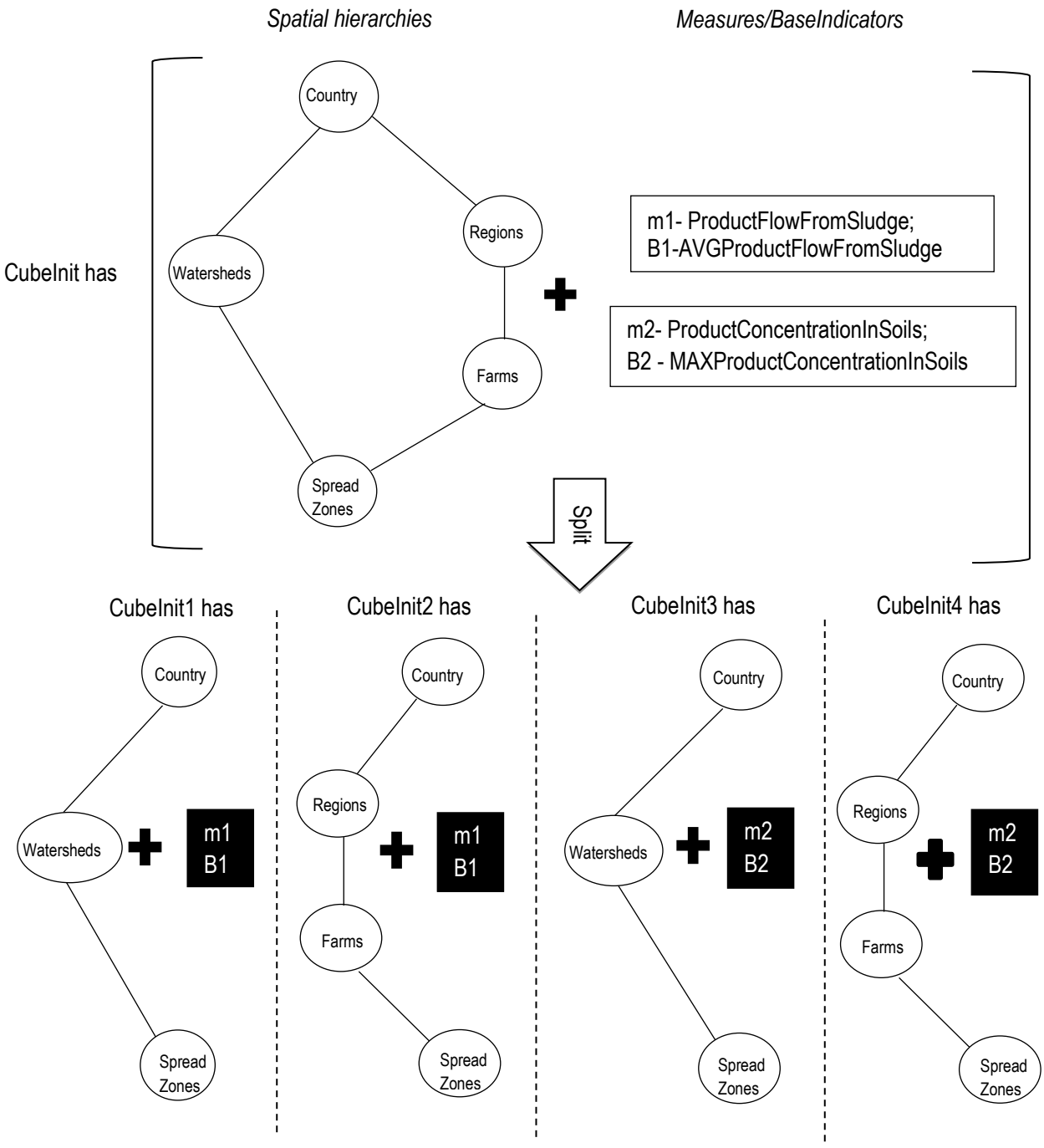


Figure 4-14: Illustration of the transformation process step 1 with our case study

Step2. In this step, some of the chosen transformation actions are applied in corresponding elementary datacube models CubeIniti by means of the transformation functions defined in section 4.5.2.1. Transformations could consist of deleting/modifying levels, aggregation rules, etc.

For example, after applying the action "Delete *SpreadZones* level" to manage m1 on appropriate CubeIniti, the resulting CubeIniti' are presented in Figure 4-15:

B1' = B1 with a new aggregation rule for Country knowing that Watershed is now the lowest level.

B1" = B1 with new aggregation rules for Regions and Country knowing that Farms is now the lowest level.

CubeInit3' and CubeInit4' are no different than CubeInit3 and CubeInit4 because the risk managed is related to m1 and not m2. The models allow users to still make analysis related to Spread Zones and measure m2.

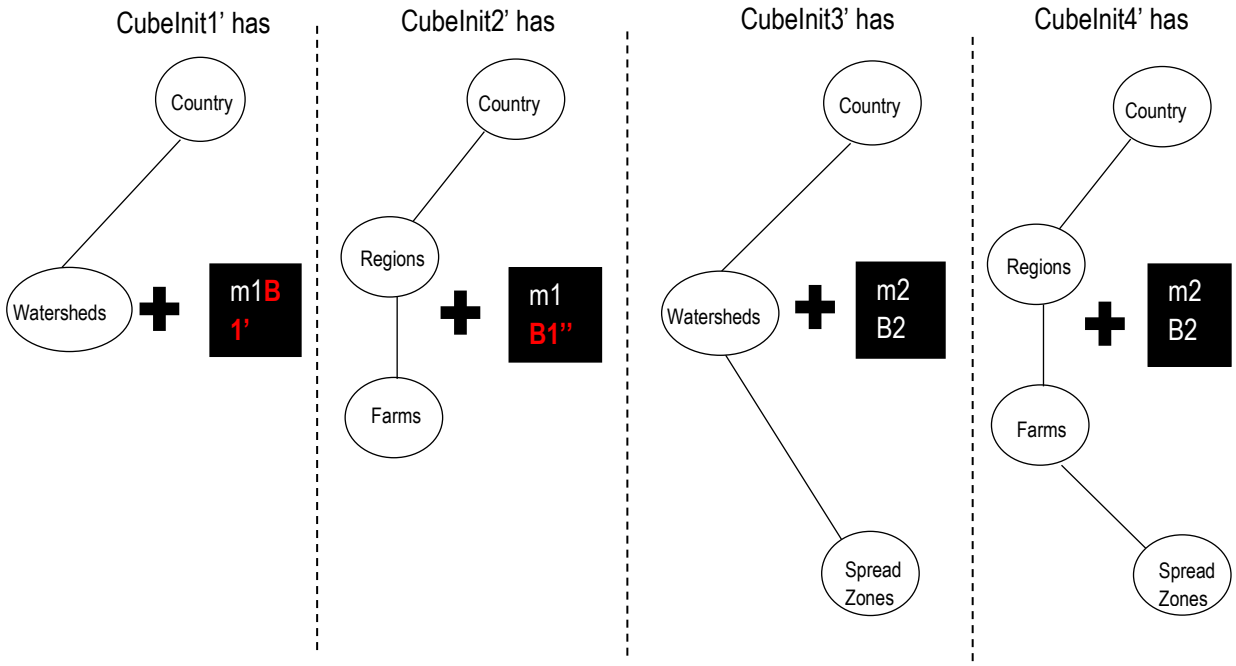


Figure 4-15: Illustration of the transformation process step 2 with our case study

Step3. At this stage of the transformation, the resulting new elementary datacubes models are merged if possible, in order to reduce the number of datacubes the users will have to analyze. The fusion of two elementary datacubes is possible only when some conditions are met (see definition 9).

Definition 9. Fusion of two elementary datacubes

*Two elementary datacube models CubeInit*i *and CubeInit*j *can be merged if:*

1. *They have the same lowest spatial level in each spatial dimension AND*
2. *They have the same couple measure/BaseIndicator.*

If one of these conditions is not respected, the two datacubes are kept separate. Ultimately, the process will deliver many datacube models, each one offering a different view of the data as required by the needs of the users in analysis.

Example 9.

For example, if we compare CubeInit1' and CubeInit2' (see Figure 4-15), we find that they do not have the same lowest level in their spatial dimension, CubeInit1' having Watersheds at lowest level and CubeInit2' having Farms. CubeInit1' and CubeIniti2' models cannot be merged. Actually, this is the case for all pairs {CubeInit1', CubeInitj'}, and {CubeInit2', CubeInitj'}, j ∈{3, 4}, so they are all kept separate. However, when comparing CubeInit3' and CubeInit4', we find that they have the same lowest level (*SpreadZones*) and the same couple measure/BaseIndicator (m2/B2); thus, they can be merged together. The process will thus deliver three datacube models for this particular transformation iteration (Cf. Figure 4-1).

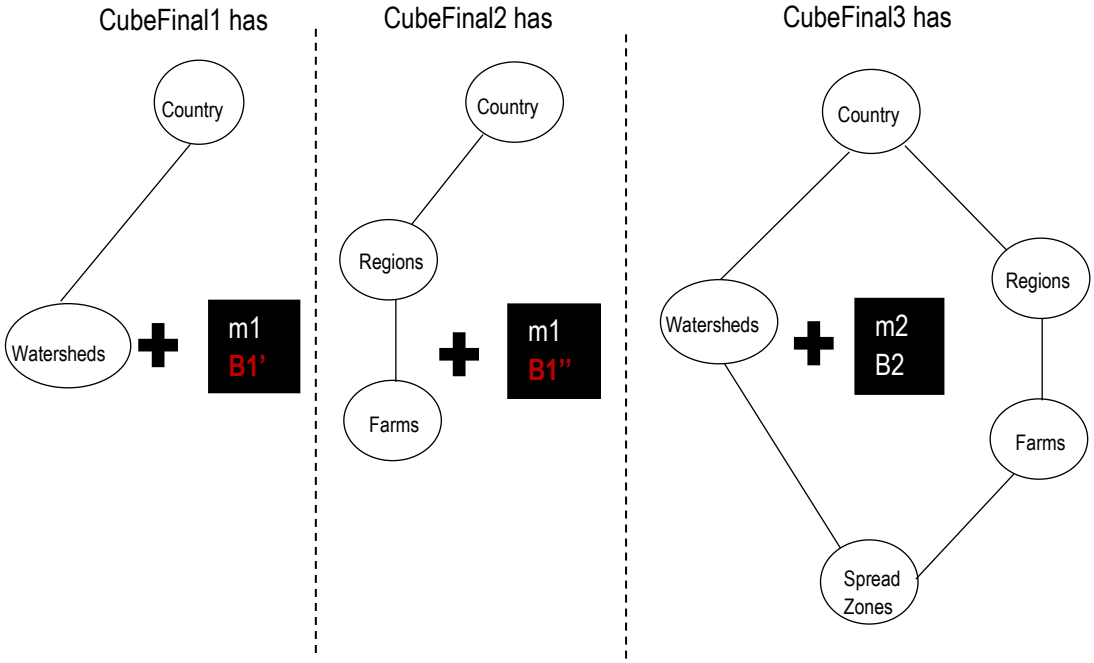


Figure 4-16: Illustration of the transformation process step 3 with our case study

4.6 Chapter synthesis

In this chapter, we have presented our new design method for conventional SOLAP applications that allows handling vague spatial data analysis issues by means of SOLAP datacube risks of misinterpretation and the tolerance levels of decision-makers to those risks. The method allows decision-makers, enterprise geospatial data and systems users and SOLAP experts to collaborate in designing SOLAP datacubes that (1) decision-makers can easily explore and analyze; (2) that can be implemented in the existing SOLAP systems; and (3) that handle the spatial vagueness on the sources. The method aim at adapting the SOLAP datacubes PIMs, to end-users tolerance levels to identified risks of misinterpretation, in an

iterative way, until end-users are satisfied with the delivered datacubes prototypes. To support the PIMs design (with vague, risks and tolerance parameters) and transformation, we have defined an UML profile, a SOLAP datacube PIMs transformation process plus we have formalized transformation functions that help apply the risks management actions chosen according to tolerance levels. The functions can be implemented in a semi-automatic design tool that will help generate and transform automatically SOLAP datacube prototypes.

Chapter 5: Risk-aware design approach evaluation

Accepted in a special issue of International Journal of Business Intelligence and Data Mining (IJBIDM), 2015
Edoh-Alove, E., Bimonte S., Bédard, Y, Pinet, F.

5.1 Introduction

In this chapter, we focus on the evaluation of the risk-aware design approach proposed in this thesis. The evaluation is about experimenting our proposals on the sludge case study presented in the previous Chapter 4 and then comparing the results obtained from our method with the classical approach results. We extracted many different SOLAP datacubes from the sludge data sources by varying the tolerance degrees and actions associated, analyzed and compared the results usability, reliability and ease of implementation with the classical approach result.

To set up our experimental framework, we have implemented and integrated our proposals into an existing CASE system that has been developed in our research team at Irstea: the ProtOLAP tool. The ProtOLAP tool is based on a ROLAP architecture and open sources OLAP server and clients. It aims at helping OLAP datacubes producers to quickly design and prototype classical datacubes by generating the logical and physical schemas based on the multidimensional model elaborated in the requirement phase. Extending the ProtOLAP tool for our implementation allows us to provide the means for a quick and semi-automatic design where the tolerance and risks parameters can be easily changed. This aspect is important because we need to be able to efficiently and quickly pass on the tolerance and risks parameters changes on the datacube schemas for many risks and different tolerance degrees and in an iterative manner. Also, integrating our new design method in such CASE system allows us to quickly verify if the produced SOLAP datacubes are easily implementable in classical SOLAP architectures. The extension essentially consisted of the implementation of the concepts of risks and tolerance degrees and the transformation functions. The programming has been entrusted to two undergrad students in their last year of Computer Science Engineer degree in the form of a 120 hours project. Their work have been co-supervised by the author of this thesis and the councilor Sandro Bimonte. The author has provided the students with the functional design of the extension, as well as the case study, the dataset and initial schema of SOLAP datacubes on which the application was tested. The supervision was organized around weekly meetings (1-2 hours) and emails exchanges. Also the author of this thesis helped in the students' final report writing by providing some of the content (project context, method global description and tool

architecture) and as a reviewer; ultimately, the author was a member of the board of examiners during the project defense.

The chapter is structured as followed: first the experimental framework built for the method evaluation is described in Section 5.2. Then Section 5.3, details the risk-aware design method validation process, presents briefly the SOLAP datacubes design and finally presents and explains the comparison results. Section 5.4 concludes this chapter.

5.2 The RADTool: an implementation of the RADSOLAP method

With the RADSOLAP method, several SOLAP datacubes are designed incrementally and can be proposed to the decision-makers. Each SOLAP datacube corresponds to a different multidimensional schema, data and visualization policies. Not only it could be difficult to perform all the required schemas transformations in a quick and coherent way that preserves the schemas validity, but it could also be difficult for decision-makers to have a good idea of the impact of all actions associated to their tolerance levels on the resulting SOLAP datacubes, their exploration and visualization, without really "playing" with them (Bimonte, Nazih et al. 2013) . For example, if one of the strategies applied is to remove a level, they may not be really sure it still fits the analyses needs until they perform the SOLAP datacube exploration. Moreover, ETL procedures are usually complex and time and resources consuming (Guimond 2005). A rapid prototyping will help them see what they can expect with the choices made, and then decide which SOLAP datacube better fits their use before moving the whole project to the costly ETL process phase.

The need of a technical solution that supports a quick design and implementation of datacubes appears undeniable. We therefore proposed the RADTool, a system to design and implement datacubes. The global architecture of the RADTool is based on the ProtOLAP system proposed in Bimonte, Nazih et al. (2013). The main idea of ProtOLAP is to allow datacube designers to automatically and incrementally implement datacube schemas and feed them with sample data, and provide end-users with real OLAP clients to allow them to test the designed datacube, in order to ultimately validate the spatio-multidimensional schema. We think that such method is highly beneficial in our risk-aware design approach.

However for now, our main purpose is not to develop a turnkey tool to support the risk-aware design but to build a system that will allow us verify that a risk-aware design approach can help produce SOLAP datacubes, exploiting spatial vague objects, where the spatial vagueness is considered (more reliable datacubes), and that are as usable and easily implementable as SOLAP datacubes designed with any classical design method. Therefore in this thesis, we concentrate on defining and implementing the RADTool features that would facilitate and make the prototyping of the datacubes quick and efficient. The

result is a first version of what could become a complete turnkey solution to support risk-aware design of SOLAP datacubes activities.

5.2.1 Preliminary work: ProtOLAP

ProtOLAP is a tool for rapid prototyping datacube development. The ProtOLAP architecture based on a Relational OLAP platform and is composed of four tiers:

- The *Requirement tier*, where OLAP experts draw a UML-based PIM, using the ICSOLAP UML profile (Boulil, Bimonte et al. 2012) and the MagicDraw CASE tool[5] (UML-based modeling software);

- The *Deployment tier*, that includes the Oracle Relational DBMS, the Mondrian OLAP server, and a mechanism that creates relational schema for Oracle and metadata for Mondrian starting from the conceptual schema;

- The *Feeding tier*, that automatically generates a visual interface through which users can feed the datacube stored in deployment tier with application domain data;

- The *Visualization tier*, that allows decision-makers to query data stored in the deployment tier using the JRubik OLAP client. SOLAP Risk-Aware Design Tool (RADTool).

5.2.2 SOLAP Risk-Aware Design Tool (RADTool)

For our RADTool, we need to extend the ProtOLAP tool with the features that are required to support the SOLAP datacube risk-aware conceptual design and prototyping activities. They are: (1) the multidimensional schemas transformations and (2) the exploration with classical SOLAP clients. As shown on Figure 5-1, the RADTool architecture is similar to the ProtOLAP one with the exception of a new *Transformation Tier* we have added.

(1) To support the schemas transformations, we have extended the ProtOLAP main interface by adding a *Risks management view* frame (see Figure 5-2); we have also added the *Transformation Tier* where some transformation functions as well as a schemas fusion module are implemented. The *Risks management view* frame is used by datacube designers to select risks and tolerance levels to add to the initial datacube. That way, the risks and tolerance levels parameters can be changed in an iterative manner and the tool can transform the datacube schemas accordingly by means of the functions (*DeleteSpatialLevel, ChangeAggregator* and *CommunicateRisk*) implemented in the *Transformation Tier.* Note that the transformations are

[5] Official Website http://www.nomagic.com/products/magicdraw.html, June 2014

applied directly on XML Mondrian schemata first generated in the *Deployment Tier* from the PIM elementary models in UML.

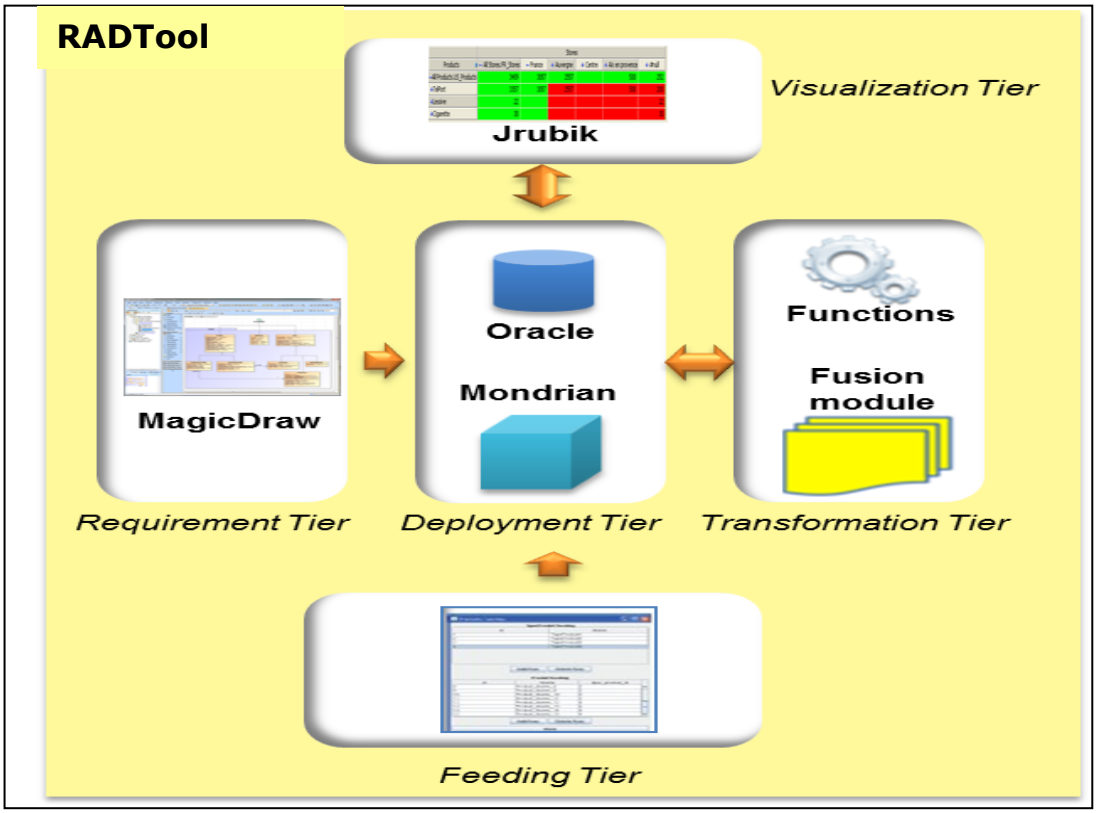


Figure 5-1: Architecture of the SOLAP RADTool

(2) In the *Visualization Tier*, the OLAP client Jrubik is delivered to decision-makers to validate the spatio-multidimensional schemas and aggregation functions as well as the choice of tolerance levels and actions (including visualization policies such as pivot cells in red or green).

An example of one datacube prototype implemented in our case study where SpreadZones level is not deleted by actions, but a visualization policy is applied, is shown on Figure 5-4. We can note that the Risk-Geometry is communicated using a red color in the pivot table as defined in the MDX generated accordingly by the tool (Cf. Figure 5-3).

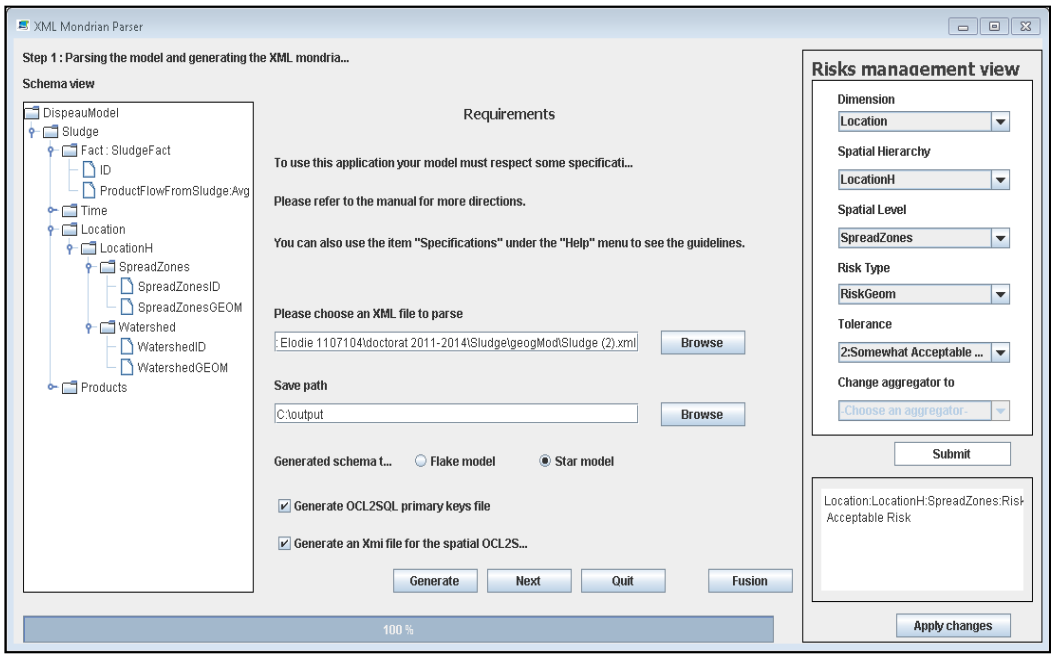


Figure 5-2: Interface for risks, tolerance levels and actions setting.

```
<Measure name="ProductFlowFromSludge_Avg" column="PRODUCTFLOWFROMSLUDGE" aggregator="avg" visible="false" formatString="Standard" />
<CalculatedMember name="Measure_RGeomComm" dimension="Measures" formula="[Measures].[ProductFlowFromSludge_Avg]">
  <CalculatedMemberProperty name="FORMAT_STRING" expression="Iif([Location].CurrentMember.Level.Name = 'SpreadZones', '|##0.00|style=red', '|##0.00|style=grey')" />
</CalculatedMember>
<CalculatedMember name="QUANTITY" dimension="Measures" visible="true" formatString="#,###.##">
  <formula>Avg(Descendants([Location].CurrentMember, [Location.LocationH].[SpreadZones]),[Measures].[ProductFlowFromSludge_Avg])</formula>
</CalculatedMember>
```

Figure 5-3: MDX styling for the visualization policy application on SpreadZones level

	Location							
Time	- France	+ Adour-Garonne	- Rhone-Mediterranee-Corse	SZ 1	SZ 2	SZ 3	SZ 4	SZ 8
-All Time.TimeHs	4,88	4,64	4,94	5,34	4,12	5,63	5,30	4,66
+2000	4,98	5,27	4,93	5,14	4,23	5,63	5,30	4,01
+2001	4,71	4,01	4,94	5,53	4,00			5,30

Figure 5-4: Pivot Table visualization in the Visualization Tier

5.3 Risk-aware design approach evaluation: process and results

To evaluate the approach, we need to verify if the SOLAP datacubes resulting from it are more reliable (spatial vagueness is considered) while remaining as usable (schema understandability) and easily implementable in classical architectures (no specific new technique proposed to support the datacubes

deployment in SDBMS and SOLAP servers) as classical SOLAP datacubes. The approach evaluation process involve activities related to this verification.

First, regarding the ease of implementation aspect, it is obvious that the implementation of the SOLAP datacubes obtained with our design method can be deployed in classical SOLAP architecture without having to develop any particular complex technique related to the spatial vagueness management. Indeed, first of all, the outputs of the RADTool are classical Spatial SQL and MDX scripts as well as XML Mondrian schemas; also, the new visual policies we have defined are simply based on the MDX styling, thus they can be enabled in any OLAP client.

The activity diagram of Figure 5-5 presents the approach adopted to verify the usability and reliability aspects.

The verification is done in four phases:

Phase 1: Define design evaluation criteria

The fundamental question we have to answer here is: “how are the SOLAP datacubes evaluated on the usability and reliability aspects?” For each evaluation activity, it is essential to define verifiable and/or quantifiable (if applicable) criteria early on to avoid any subjectivity. This step is where the criteria definition is done according to our goals.

Phase 2: Design SOLAP datacubes using RADSOLAP method and design SOLAP datacubes using a classical method

This phase simply focuses on the designing and prototyping of the SOLAP datacubes with and without our new risk-aware design method.

Phase 3: Analyzing resulting datacubes based on the criteria

In this phase, we evaluate the design results using the criteria defined in the first phase.

Phase 4: Discuss analysis results

In this step, a comparison is done between the results of the classical design and the RADSOLAP design based on the previous analysis. In this phase, we are able to answer the question “Are the SOLAP datacubes designed with our method more reliable than and as usable as the classical datacubes?”

In the rest of the section, we detail each step and results obtained.

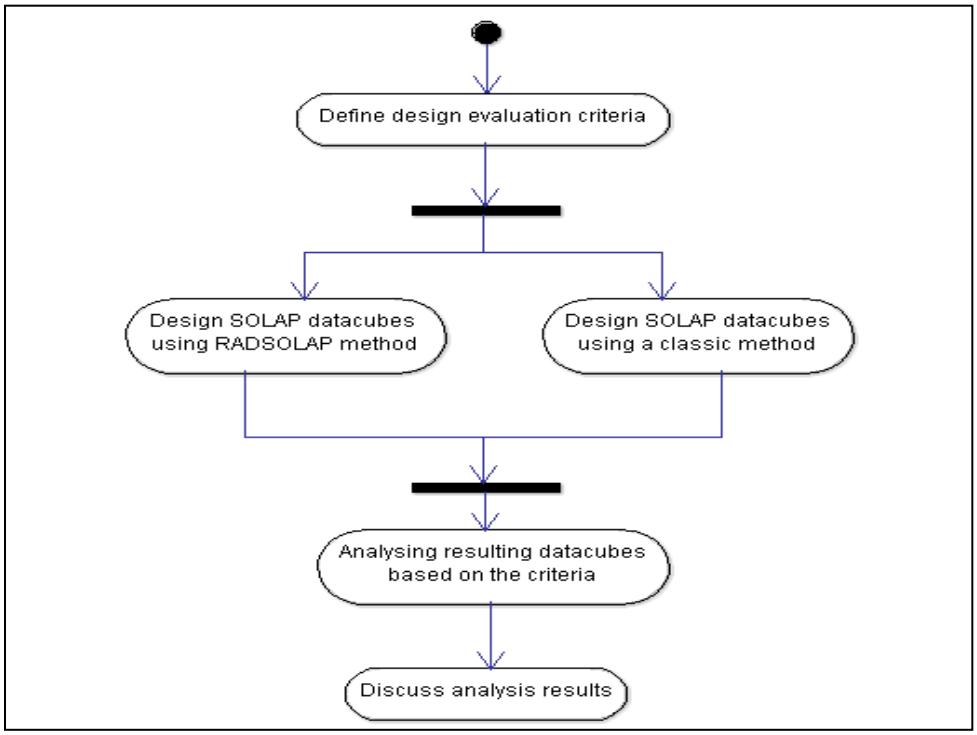


Figure 5-5 Datacubes usability, reliability and ease of implementation verification process

5.3.1 Definition of the design evaluation criteria

5.3.1.1 Usability evaluation

Some researchers have advocated the use of quantifiable metrics to test datacubes conceptual or logical schema understandability (Berenguer, Romero et al. 2005, Serrano, Trujillo et al. 2007, Golfarelli and Rizzi 2011). Valid and useful metrics definition is a complex activity. It requires a clear definition of the measurements goals and organization's needs, then the definition of the metrics based on the goals and needs, and finally the most important step, the metrics validation (Serrano, Trujillo et al. 2007). The validation should be theoretical and empirical (based on experiments, case studies and surveys). This thesis not being about defining new metrics, we find judicious to identify appropriate already existing metrics to use for our datacube understandability evaluation.

The two main requirements that guide our metrics choice are: (1) facts, dimensions and measures must be considered and (2) the risk communication artifacts must be addressed.

Ultimately, we have selected the following metrics: *Total number of classes, Number of spatial dimensions, Number of hierarchy relationships, Number of measures per fact* as defined in Serrano, Trujillo et al. (2007) and *Number of multiple hierarchies* as defined in Gosain, Nagpal et al. (2011).

- *Total number of classes*: it counts the total number of classes existing in the SOLAP datacube PIM. The classes are the level classes, the fact class, and the BaseIndicator (Boulil, Bimonte et al. 2012) classes. This metric in particular allows us to consider a new risk class, to hold the risks, tolerance and visualization policies variables associated, introduced by our method into the SOLAP datacube models.

- *Number of spatial dimensions*: it counts the number of spatial dimensions existing in the SOLAP datacube PIM. We have selected this metric because it allows us to evaluate the complexity of the model regarding the number of spatial dimensions end-users will have to wrap their mind around and exploit, with or without vagueness issues, in their analysis.

- *Number of hierarchy relationships*: it counts the number of relationships between the levels in a dimension. It highlights the importance of complex hierarchies (Malinowski and Zimányi 2008) in the model. We have selected this metric because it allows evaluating the understandability of the model.

- *Number of multiple hierarchies*: it counts each time a multiple hierarchy is found in the model. We have selected this metric because it help evaluates and compare the models complexity regarding the number of multiple hierarchies end-users will have to deal with while exploring the datacubes.

- *Number of measures per fact*: it counts the number of measures associated to each fact. This metric focuses on the measures described in the model and we found it useful in this work because it allows us to evaluates and compare the models complexity on the number of measures end-users will have to exploit with or without spatial vagueness issues.

Using these metrics as criteria for comparing the datacubes allow us to objectively verify if the RADSOLAP datacubes schemas are as usable as classical SOLAP datacubes ones. How do the metrics values differ from one datacube to the other? Are the values greater or lesser for the RADSOLAP datacubes? Those are the questions we want to answer during the phase 4 of the validation. Indeed, if the metrics values are similar, we could conclude that the method keeps the level of usability of the datacubes. However, if they differ in a significant way (e.g. 30% more or less), we could evaluate if the RADSOLAP method produces more usable datacubes or not in this study case:

- A *total number of classes* that has increased in a significant way for RADSOLAP datacubes would mean that the method adds too many new risk classes for the datacubes to be as understandable as the classical datacubes.

- A *Number of spatial dimensions, hierarchy relationships*, *measures per fact* or *multiple hierarchies* significantly greater would mean that the RADSOLAP datacubes are less understandable.

In contrary, a smaller number of classes, spatial dimensions, hierarchy relationships, measures per fact or multiple hierarchies would mean that the method simplifies the datacubes and makes them more usable in regards with the considered criterion.

5.3.1.2 Reliability evaluation

In this work, reliability refers to the fact that the spatial vagueness is considered (identified and managed) or not in the datacube. The question of how the reliability is evaluated has been the main issue. Indeed, this concept being very particular to our work, we could not find any existing metric or criteria suitable for this evaluation activity in the literature.

With that said, in our RADSOLAP method (Cf. Chapter 4, Section 4.5), more specifically at step 2, we advocate the identification of vague geometric attributes on the initial SOLAP datacube PIM and then at step 3 the identification and assessment of all the risks of misinterpretation induced by the previous identified vague geometric attributes. The risks identification is then followed by risks management activities in the next step. In summary, each vague geometric attribute induces zero to multiple intrinsic risks of misinterpretation (risk-geometry and risk-aggregations), and each risk of misinterpretation is managed with 0 to multiple risk management actions (Cf. Figure 5-6).

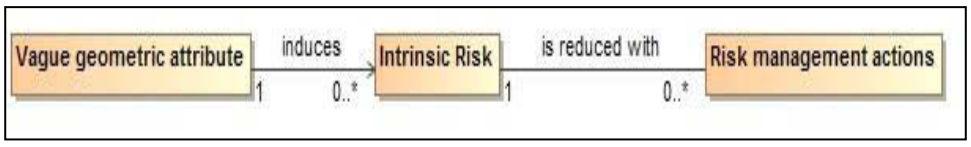


Figure 5-6: Relationship between vagueness, risks and risk management actions

Basically, the method allows considering the spatial vagueness. The spatial vagueness management is however replaced by the management of the risks of misinterpretation (induced by the vagueness). Any effort towards considering the risks of misinterpretation, namely a risk identification activity, and any effort towards managing the eventual identified risks, namely a risk management action is an effort towards managing the spatial vagueness and must be recognized as such.

Ultimately, to evaluate the reliability for both classical and RADSOLAP datacubes, we just need to verify if the spatial vagueness has been highlighted and risks have been identified and managed or not.

5.3.2 SOLAP datacubes classical and RADSOLAP design

Classical design:

The classical design method result is a SOLAP datacube PIM where the spatial vagueness is not considered at all (see Figure 5-7). It is the same as the **Sludge** PIM showed on Figure 3-6 except that the spatial dimension is enriched with new spatial levels and a new measure to meet more user analysis requirements. It has now one multiple hierarchy with two paths as shown on Figure 5-7: Spread Zones < Watersheds < Country (*LocationHWatershed*) and Spread Zones < Farms < Regions < Country (*LocationHFarms*). The new measure, *ProductConcentrationInSoils* refers to the trace metals concentration in the soils. It is also important to monitor this concentration in order to control the quantity of product added by the spreading. The new path *LocationHFarms* allows end-users to monitor the product flow brought by the sludge spread and the product concentration in the soils not only in spread zones and watersheds but also in farms and regions.

RADSOLAP design:

We have considered the spread zones in their minimal extents (Cf. green region on Figure 4-3). We note that considering them in their maximal extents (rather than their minimal extents) will not change the results of our analysis. The only things that change when exploiting the maximal extents in this case study is the wording of the risks identified (over-evaluation became under-evaluation, vice-versa or nothing changes) and eventually end-users tolerance levels since the risks are different. Since we will be trying all possible combinations of tolerance levels here, those changes do not impact the design resulting and consequently the results evaluation.

At first, we built an initial PIM where all vague data and risks of misinterpretation are identified. Then we split that initial SOLAP datacube into elementary SOLAP datacubes (Cf. Appendix E for the PIMs obtained). We have applied a combination of all 4 tolerance levels to the elementary PIMs and actions were applied in accordance with the tolerance choices. The result is a set of 31 final SOLAP datacubes PIMs, which are actually all possible datacubes for this study case.

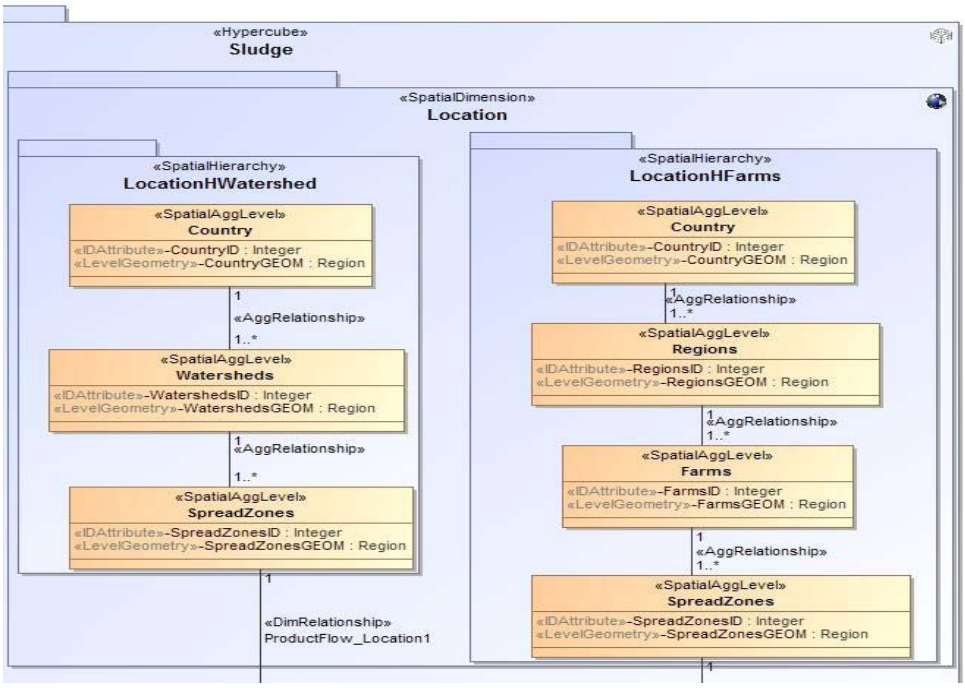


Figure 5-7: Classical Sludge datacube PIM new spatial dimension

5.3.3 SOLAP datacubes reliability and usability comparison

Reliability:

While the Sludge datacube has been designed without considering, let alone managing the spatial vagueness on the spread zones, the spatial vagueness on spread zones has been singled out and taken care of by means of risks of misinterpretation identification and management for the 31 RADSOLAP datacubes. In some cases, the choice to accept some of the risks, and therefore not doing any action to reduce them has been made; however, that choice is conscious and is part of a risk management strategy (indifference) itself.

We can conclude that the SOLAP datacubes designed with the RADSOLAP method are more reliable than the classical Sludge datacube.

Usability:

Now, regarding the usability aspect, the SOLAP datacubes PIMs were tested according to the metrics identified previously. We recall that we have 31 final possible SOLAP datacubes for when the spread zones are considered in their minimal extents and 1 classical SOLAP datacube.

First we have computed the Euclidian distance d between the vectors of metrics values corresponding to each resulting SOLAP datacube and the Sludge one. This distance will be used to evaluate the level of

similarity between the RADSOLAP datacubes and the classical and help us answer our first question: How do the metrics values differ from one datacube to the other?

We consider:

$V(x_1, x_2, x_3, x_4, x_5)$ the vector of metrics representing the *Sludge* datacube,

$V_{risk}(y_1, y_2, y_3, y_4, y_5)$ the vector of metrics representing one of the 31 SOLAP datacubes obtained with the RADSOLAP method.

$d = \sqrt{\sum_{i=1}^{5}(x_i - y_i)^2}$ the Euclidian distance . The computed distances are available on the Figure 5-8.

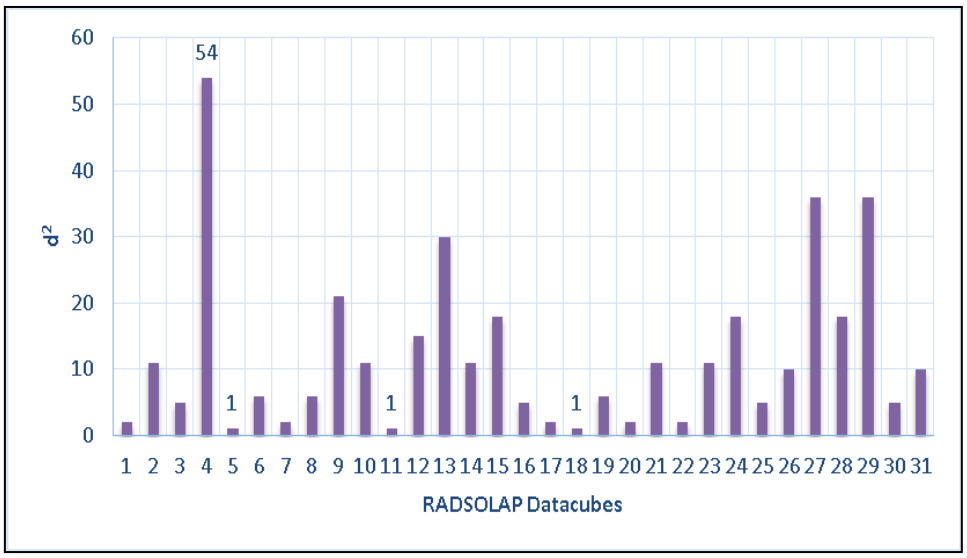


Figure 5-8: Figure showing the distance² values for the 31 RADSOLAP datacubes

Overall, three RADSOLAP datacubes PIMs are at a distance of 1 from the classical Sludge PIM, only one is at a distance of 54 (greatest distance) and the majority is around the median distance (d^2=12). The set standard deviation is 12.21. Figure 5-9 detailed the statistics for each criteria. It shows that the metrics values are whether the same for all SOLAP datacubes, include the datacube Sludge, or that the metrics values for the datacubes produced with the RADSOLAP method are lesser than the ones for the classical SOLAP datacube. It is especially the case for the metrics *Total number of classes, Number of hierarchy relationships*, *Number of multiple hierarchies*. We can already conclude that the majority of the SOLAP datacubes designed with our method are similar to the classical **Sludge** datacube in terms of the usability.

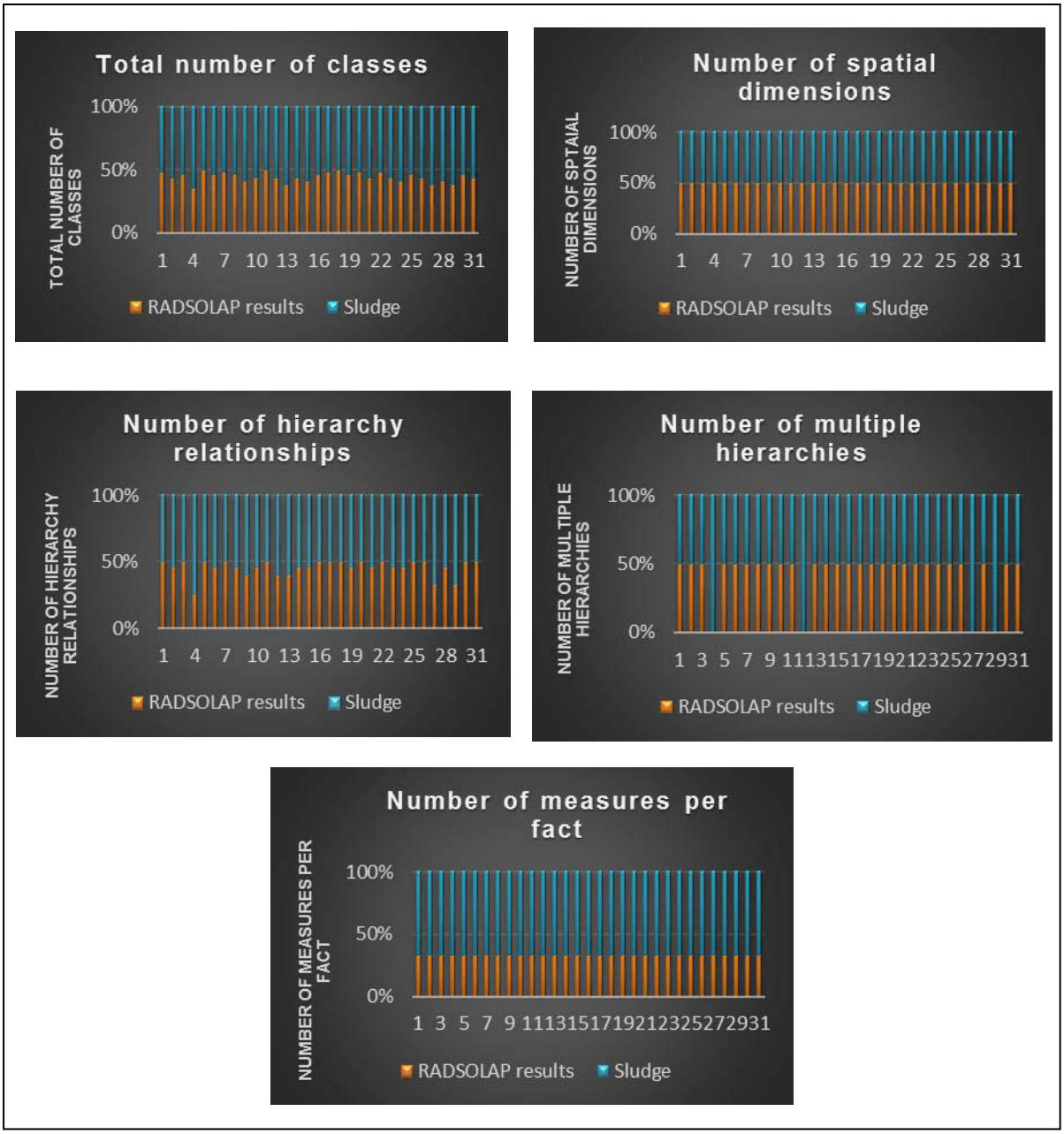


Figure 5-9: Diagrams illustrating the comparison of the values

To refine the comparison, and answer our second question ("Are the values greater or lesser for the RADSOLAP datacubes?") we take a closer look to the RADSOLAP datacubes PIMs corresponding to the smallest, greatest and median distance. We have selected three SOLAP datacubes from the 31 designed: one that corresponds to the smallest d^2 value (*SludgeRiskSmallest*), one that corresponds to the highest d^2 value (*SludgeRiskGreatest*) and one that corresponds to the median d^2 value (*SludgeRiskMedian*). The metrics values for the **Sludge** datacube and the ones for the three selected SOLAP datacubes are computed in the following Table 5-1.

	SOLAP datacubes			
Metrics	**Sludge**	**SludgeRisk Smallest**	**SludgeRisk Greatest**	**SludgeRisk Median**
Total number of classes	13	13	7	10
Number of spatial dimensions	1	1	1	1
Number of hierarchy relationships	6	6	2	5
Number of multiple hierarchies	1	1	0	1
Number of measures per fact	2	1	1	1

Table 5-1 SOLAP datacube PIMs usability testing results

For each one of the three RADSOLAP datacubes the metrics values are whether the same as the Sludge datacube metric values, or lesser. It is especially the case for the metrics *Total number of classes, Number of hierarchy relationships* and *Number of measures per fact.*

Regarding the *SludgeRiskGreatest* PIM in particular, even though its classes include new classes holding risks identified + tolerance +visualization policies, it appears that this datacube is still more usable than the classical Sludge one (only 7 classes, 2 hierarchy relationships and 0 multiple hierarchies).

Also, we think that the visualization policies themselves, which are new artifacts brought by our approach, do not harm the usability since in the worst case, only three of them are to be included in the datacubes interpretation by the end-users. Indeed, we note that in general, the RADSOLAP datacubes only holds 0 to at most 3 visualization policies variables to communicate the risks when applicable.

With all these observations, **we have come to the conclusion that each SOLAP datacube designed with the RADSOLAP method for this case study is as usable (or even more usable) as the classical SOLAP datacube designed.**

5.4 Chapter synthesis

In this chapter we evaluated the risk-aware design approach. At first we presented the RADTool, a CASE SYSTEM that supports the risk-aware design method. The RADTool helps generate automatically SOLAP datacubes prototypes that end-users can visualize and explore in order to validate the whole design. The RADTool has been used specifically to produce different SOLAP datacubes corresponding to all possible combinations of tolerance levels for our case study based on the French National sludge spread monitoring database (Soulignac, Barnabé et al. 2006). Then, we analyzed the SOLAP datacube design with the RADSOLAP method as well as the one obtained from a classical design approach in terms of

reliability, usability and ease of implementation in classical architectures. The analysis is based on quantitative criteria (total number of classes, number of multiple hierarchies, number of spatial dimension, number of hierarchy relationships and number of measures per fact) defined beforehand for the usability in particular. The analysis has shown that, overall and for this case study, our risk-aware approach provides SOLAP datacubes were the spatial vagueness is considered (more reliable), that are as understandable and as easily implementable in classical SOLAP architectures as classical SOLAP datacubes.

The question that remains unanswered is: “How the usability is impacted when there are more than one datacube provided to the end-users, knowing that the process can deliver from 1 to N (N being the number of elementary datacubes) SOLAP datacubes?” To answer this one, we think that it is necessary to place the usability testing in the context of multiple datacubes exploitation, however for now, to the best of our knowledge, there are not objective methods or quantitative approaches to such testing in the literature yet. We will address this question more in details in our future work.

Chapter 6: Conclusion and research perspectives

6.1 Introduction

As stated before, one of the challenges when dealing with spatial vague data in SOLAP datacubes is to take into account the spatial vagueness on spatial data while maintaining the usability and the ease of implementation of the datacubes. In this thesis, we wanted to progress in this direction by adopting a new symbiotic approach for the design of the SOLAP datacubes. Our main contribution is the fundamentals for an innovative SOLAP datacubes design method that is based on the introduction of risk management steps into the classical hybrid design method. This method is very particular in three different ways: it allows the identification of potential risks of misinterpretation at an early stage of the datacubes design, it allows taking user's tolerance levels to the risks identified into account during the design, and most importantly, those tolerance levels are used to trigger and guide an unprecedented dynamic datacube transformation process that is the core of the risk-awareness we want to achieve.

In **Chapter 1** of this thesis, the research context was presented alongside with a targeted literature review from which the problem was defined. In fact, either the spatial vagueness is neglected in spatial databases, whether transactional or multidimensional, or it is handled by representing the spatial objects with vague objects models such as fuzzy or exact models. In the latter case, not only a practical and efficient integration of the vague objects models in existing SOLAP technologies is yet to be achieved, but also dealing with vague objects models during the decision making process can be a disadvantage in many contexts of application. The majority of end-users (which are decision-makers) are used to the crisp point, line and polygon representation of geographic phenomenon and changing that to a complex fuzzy model or exact model calls for a different and complex way to visualize and analyze results (more data, more complex representation and aggregations methods). In some contexts of application, the complex vague objects can be the best choice to have accurate models and results (for example in Hazards modeling and multidimensional analysis), but in other contexts, the end-users might end-up having highly accurate spatial data but not really usable datacubes due to the amount of data they have to analyze. It can be for instance the same hazards analysis case but in the context of damage costs analysis over the years and for different regions and activity sectors. We then stated our research question which concentrate on whether it is possible to adopt a new approach, based on a symbiotic trade-off between spatial data accuracy, usability and ease of implementation in existing SOLAP technologies, to produce relevant SOLAP datacubes. This led us to define the thesis main objective: proposing a SOLAP datacubes

design method integrating a risk of misinterpretation management method to deal with spatial vagueness in a symbiotic trade-off approach. Specific objectives and a research methodology were ultimately presented in this chapter. In **Chapter 2**, we study the relevant concepts underlying the research areas the problem is related to. It is the notions of spatial data uncertainty and spatial data quality which is the core of the research problem; the main concepts related to Spatial OLAP systems that we needed to understand in order to analyze the effects of the presence of spatial vagueness in datacubes on the analyzes, alongside with the way those datacubes are classically designed and implemented in theory and in practice; and finally the fundamentals of risks management that are needed to develop the risk-awareness aspect of our design method. **Chapter 3** is where we started addressing our problem by taking on the first sub-objective which lays in proposing a risk-aware SOLAP datacubes design approach. The proposals developed here were used to elaborate a rapid prototyping SOLAP datacubes design method in the next chapter (**Chapter 4**), allowing us to address the second sub-objective of the thesis. The third and fourth sub-objectives were the focus of the **Chapter 5** where an experimentation of the approach was done on an agricultural sewage sludge spreading case study using a CASE system implementing the method core. The purpose of the experimentation was to validate the method by doing a comparison between all possible SOLAP datacubes designed with our new method to a classic SOLAP datacube in terms of usability, reliability and ease of implementation in classical SOLAP architectures.

In the following sections, we describe and discuss the main contributions of this research project regarding each sub-objective (Section 6.2) and then some research perspectives are provided (Section 6.3).

6.2 Contributions and discussion

6.2.1 Sub-Objective 1: To propose the fundamentals of a risk-aware SOLAP datacubes design approach

The literature review revealed the lack of a design process combining users-driven and sources-driven approaches in addition with a risk management method in the computer science domain. Also, the existing SOLAP datacube design methods that allow risks management in Geomatics can be improved by developing risks identification and control tools specific to spatial vagueness. Accordingly, the main contribution regarding this sub-objective was a complete step by step hybrid SOLAP datacubes design process where steps of risks identification, assessment and management according to end-user tolerance levels are integrated to enable the risk-awareness. This risk-aware design process is composed of two main phases that are the requirements specification phase and the conceptual design phase. The main purpose of proposing this new process being to allow the integration of risk-aware steps into the classical hybrid process, we have added steps of vagueness and induced risks identification to the requirement specification phase which comprises also the common users and data sources identification as well as user's analysis requirements identification. To the conceptual design phase, we have added a step of risk identification that has to come after the datacube initial design, and steps of risks assessment via the

expression of the tolerance levels, risks management by means of actions to be executed on the SOLAP datacube design, and SOLAP datacube schema transformations according to the actions chosen previously, in that order.

This process is quite generic and can be implemented in another uncertainty (or quality) and risk management context and with different risk management tools by changing the uncertainty/quality element (e.g. Managing the incompleteness and risks of inappropriate use associated). However, because the spatial vagueness and relative risks assessment and management is the core of the risk-awareness in this thesis, it was important to address them specifically in this thesis. Thus, we first characterized the spatial vague data in a simplified way that complies with our symbiotic trade-off vision. Then we qualified the risk of misinterpretation before analyzing in which forms such risks are expressed along the hierarchies of a SOLAP datacube exploiting spatial vague data. This results into the definition of the concept of risk of misinterpretation of SOLAP datacubes and its classification in two main categories: intrinsic and extrinsic risks. Those contributions are to be exploited as advocated by the design process to identify vagueness and risks. Finally, we provided a tolerance scale and corresponding risk management strategies and actions to support the risks assessment and management advocated by the new design process.

Ultimately, applying the whole process to a simplified spreading use case allows us to validate the pertinence and feasibility of each contribution. From that, we can say that the sub-objective was achieved.

Discussion:

- *Risks of misinterpretation definition and classification*

For this classification, and for the whole thesis, we focused on spatial vector data, especially polygons. The classification can still be applied in case we are dealing with points or lines but not without an extension of the intrinsic risk category. The extension can be in the form of new risk classes that will compose the risk-level. It is even possible to apply the classification to other types of risks of datacube inappropriate use in general. There will always be intrinsic and extrinsic risks of datacube inappropriate use. However, determining all types of risks of inappropriate use can be a thesis topic in itself, with the objective of elaborating a risk ontology.

- *Risk-aware design process*

This design process is very innovative but stick with the classical design approach at the same time. Its main distinctive characteristic is the iterative SOLAP datacubes schemas transformation performed before even reaching the first accomplished model (one that suits the initial end-users' requirements). Revising datacubes structures based on identified related risks has never been done before. Usually a model is directly drawn out from the requirements and then the designers can go back to revise that model relatively to new inputs (coming from users, application context or data sources).

This process can be adapted to other quality or uncertainty issues (incompleteness, temporal uncertainty, fuzziness on qualitative attributes etc.), for transactional databases design and/or with different risk management tools (risk ontology, risk identification descriptive files etc.).

- *Risks assessment and management*

The main advantage of the risk assessment and management tool provided is its simplicity of use. Indeed the choice of a scale of four level of tolerance was dictated by the need to keep the scale as simple as possible for the design process to also stay simple and easy for datacube producers. Also by making an equivalence between the tolerance scale and the risk management strategies, it makes it easier for SOLAP experts to convey the end-users tolerance into risks management actions, whether it is done manually or automatically. With the addition of the risk management actions categories, they can focus more on the design (their expertise) instead of the risk management complexity.

Other advantages lie in the fact that the categorization of possible risk management actions allows the addition of new actions in the future, and the fact that the association between tolerance level and strategy can be rearranged differently if needed. This will not change anything for the design process defined previously. Therefore, we can say that risk assessment and management tool provided is quite scalable.

6.2.2 Sub-objective 2. To propose the principles of a practical implementation of the risk-aware approach

First, we translated the new risk-aware approach into an agile design process that would walk the datacubes project committee through the SOLAP datacubes description (vague, risks and multidimensional structure), the structure transformation according to the risk-aware approach and finally the SOLAP datacubes design validation: it resulted in the risk-aware rapid prototyping method (RADSOLAP method). The process steps go from the users' requirement specification all the way to the datacube prototyping and final delivering, through the vagueness and risks identification, the tolerance levels expression by users and datacube PIM transformation according to users' tolerance levels.

To help insure the final datacubes PIM delivered are well formed and all the tolerance parameters were computed in the right way, we have worked out an UML profile for the PIM elaboration (RADSOLAP UML Profile), in addition with a formal definition of PIMs transformation functions and a transformation process. The UML profile is an extension of the ICSOLAP UML profile developed by Kamal Boulil in the context of his PhD thesis. The choice of this ICSOLAP UML profile for our method is mainly motivated by the fact that the profile has been implemented in practice, making the automation of the datacube PIMs transformation possible. The second reason behind that choice is that the profile belongs to our research team and it is interesting for us to take it further with our contributions. The transformation functions formalization is supported by the RADSOLAP UML profile; the functions are destined for the application of the risk management actions, proposed in Chapter 3, on the SOLAP datacube schemas. As for the

transformation process, it aims at describing how a SOLAP datacube PIM can be transformed during the design in a way that conserve its validity.

The contributions were tested manually on the sewage sludge spreading case study (SILLAGE) progressively with successful outcomes. With that, we were able to validate our sub-objective.

Discussion:

- *A risk-aware rapid prototyping design method*

Our rapid prototyping risk-aware design method presents the following advantages:

- It allows the exploitation of classical crisp geometries
- It allows taking into account any spatial vagueness issue on the data sources, in contrary to most of the methods proposed in the computer science.
- It allows involving the end-users themselves not only in the way the spatial vagueness issues are managed but also in the validation of the design itself by letting them play with prototypes.
- The delivered datacubes implementation is identical to the implementation of any classical SOLAP datacube.

- *An UML Profile for SOLAP datacubes PIM design taking into account the new artefacts that are the risk classes, tolerance values and risk communication variables*

There are many benefits to the UML Profile in this context as stated previously in this thesis. The ones are the most valuable for us are the fact that it empower SOLAP experts with a visual formal tool to well describe the datacube with the new paradigms that are the vagueness, the risks, the tolerance levels and risk-communication policies and the fact that it makes possible a semi-automatic design of the SOLAP datacubes schemas. Such profile have never been proposed before and we believe that it is a very useful contribution for the combination of spatial data quality and design methodologies research fields.

- *A SOLAP datacubes PIM transformation functions formal definition*

The functions are designed to be primitive in a way that each one of them execute a single part of a transformation action. They can be combined to perform a whole action or even more complex ones. Being based on our UML profile, they can be implemented in any tool as long as the profile is adopted. Otherwise, it an adaptation of the definitions to the spatio-multidimensional model to be used is required.

- *A SOLAP datacubes PIM transformation process (Split-Transform-Merge)*

The transformation process can seem complex to be applied manually. However, it at least guarantees the SOLAP datacube schema logical coherence and validity throughout the design, whereas, applying different management actions on an initial schema according to different tolerance levels can be very painful, confusing and subject to various errors due to the designers inattention or to the incompatibility between some of the actions chosen. With that said, in this thesis, we did not specifically address the order in which one should apply the changes to make the task smarter. For instance, it does not make sense to apply an action of attaching a risk communication policy to a certain spatial level before moving on with deleting the inferior spatial level (tolerance 0 on a risk-geometry on that level) knowing that the risk identified at that superior level comes directly from the uncertainty on the level to be deleted. A priority order to the actions application must be thought trough, tested and validated to complete this process.

6.2.3 Sub-objective 3: To demonstrate the implementation feasibility of the proposed risk-aware approach.

One of the gaps identified in the literature regarding spatial vague objects management is the absence of technical implementation tools for the proposals provided. This thesis having also a strong foot in the Computer Science domain, it was important for us to demonstrate the technical implementation feasibility of our proposals; more specifically, we needed to test the implementation in a rapid prototyping CASE system (ProtOLAP) we have worked on with our Irstea research team (Bimonte, Nazih et al. 2013). To achieve this sub-objective, we have extended ProtOLAP using the principles advocated in sub-objective 1 and 2. We mainly developed and added a *Transformation Tier* to the ProtOLAP system, in addition with a risk management graphical interface that SOLAP datacube producers can use to key in vague, risk, and tolerance parameters to the initial SOLAP datacube multidimensional schema. In the new Transformation Tier, some SOLAP datacube transformation functions were made available as well as a Fusion module for datacube schemas merger as advocated by the RADSOLAP method. The result, called the Risk-Aware Design Tool (RADTool) was tested on our sewage sludge case study with positives results. Indeed, we were able to design the SOLAP datacube using our RADSOLAP UML profile, transform the schema, based on our tolerance scale and corresponding actions, using our transformation functions, and finally implement the resulting schema in a classical architecture (Oracle + Mondrian + Jrubik).

Discussion:

Being designed for OLAP datacubes prototyping, ProtOLAP was not destined to manage the spatial features. The Deployment Tier does not support spatial objects storage and spatial predicates, the *Feeding Tier* does not support spatial data insertion and the *Analysis Tier* does not allow a map visualization and exploration of the data. Also for now, a complete real Spatial OLAP system based on free solutions (like Jrubik and GeoMondrian) is not possible without doing a programming work on the

client module in consequence (Bédard, Proulx et al. 2005, Malinowski 2014). Because such work is time consuming, and not essential for our demonstration, it was left out of this thesis. Indeed, the XML and SQL schemas being generated for datacubes exploiting single polygons as geometric data, there is no reason why those schemas and the spatial data to be exploited can't be handled also with classical SOLAP servers' operators, spatial ETL tools, spatial DBMS and SOLAP client regarding the deployment, feeding and exploration of the SOLAP datacubes. In consequence, the current version of the SOLAP RADTool does not allow the management of the map visualization and exploration.

The only thing is that this limit didn't allow an eventual demonstration of the implementation feasibility of risk communication policies for the map visualization. But as we can see, proposing those policies was not one of our objectives, even though we think that such proposition would be also valuable for the Geomatics aspect of this research topic and therefore be considered for future research.

6.2.4 Sub-objective 4: To demonstrate the benefits of the risk-aware approach.

This sub-objective calls for the approach evaluation on all the criteria we have chosen to concentrate on in this thesis: the datacube schema understandability, their ease of implementation and their reliability. We tested a total of 31 SOLAP datacubes designed with our RADSOLAP method for the sewage sludge case study. Those datacubes are the results of the application of all combinations of tolerance levels for all the risks identified on the initial SOLAP datacube schema.

The datacubes are all implementable in existing classical tools i.e. PostgreSQL/PostGIS database, Mondrian Server and JRubik client just as any classical SOLAP datacube. Their improved reliability were demonstrated with success based on the direct relation we established between vagueness consideration and risk management. Regarding the schemas understandability, we have compared the 31 SOLAP datacubes with the one designed in the classical way based on the following metrics: Total number of classes, Number of spatial dimension, Number of multiple hierarchies, Number of measures per fact and Number of hierarchy relationships. It appeared that the metrics values are in most cases the same for all compared SOLAP datacubes schemas; In all other cases, the values are even inferior to the ones computed for the classical SOLAP datacube schema (Specifically for the Total number of classes, Number of hierarchy relationships and number of multiple hierarchies).

The risk-aware design approach definitely helped designed more reliable and usable SOLAP datacubes when spatial vagueness is involved. The only downside is that in our method, sometimes the end-users will be provided with more than just one final SOLAP datacube, depending on the merger result; also the datacubes have new classes holding risks identified in addition with the end-users tolerance and visualization policies variables when applied. Regarding the latter, the schema understandability is still kept or better than the classical SOLAP datacube understandability, in our case study; indeed as we found out, the number of classes, hierarchy relationships or multiple hierarchies are lower while the other criteria

values are still the same for both type of datacubes. We have also noted that the presence of new artifacts that are the visualization policies variables in the multidimensional schema does not jeopardize the usability; indeed, only three variables at most are added to each SOLAP datacube to communicate the risks when applicable.

Discussion:

This thesis proposed a solution that have never been mentioned before to a recurrent and old issue: spatial vagueness management. It replaces the spatial vagueness management by a risk of misinterpretation management based on a compromise between the theoretical accuracy regarding spatial vagueness, usability and ease of implementation of SOLAP datacubes exploiting spatial vague objects.

We recall that our thesis hypothesis is that **a design approach based on a symbiotic trade-off vision allows producing SOLAP datacubes, exploiting spatial vague objects, that remain as usable, reliable and easily implementable as classical SOLAP datacubes.**

We believe that this hypothesis was validated by (1) proposing a new risk-aware design approach, (2) proposing a prototyping design method based on that approach and (3) implementing, testing and evaluating the approach on our case study.

The two questions we still want to answer regarding the usability testing are: (1) How the usability is impacted when there are more than one datacube provided to the end-users, knowing that the process can deliver from 1 to N (N being the number of elementary datacubes) SOLAP datacube?, and (2) How the front-end usability is impacted by the added visual parameters that are the risks communication policies?

For the first question, it is necessary to place the usability testing in the context of multiple datacubes exploitation and propose, validate and apply methods and quantitative approaches appropriate for such testing. The second question on the other hand requires a complete experimentation of the RADSOLAP method on a real case study, with delivery of physical prototypes, fed with real data and explored by different end-users. Since the proposals underlying our RADSOLAP method focused on the multidimensional structure for the most part (UML profile, schemas transformation functions and process), the schemas usability testing was sufficient for this thesis; therefore, we think that the frond-end usability can be done as a future research.

The effectiveness of the risk-aware approach basically relies upon the risks of misinterpretation identification. Indeed, when all the risks are well identified, it is possible to take them into account and thereby take the spatial vagueness into account. The percentage of risks identified as well as the percentage of risks managed have an impact on the SOLAP datacubes reliability per definition. The

reliability level can thus be described by the combination of the risk identification level and the risk management level. We have three interesting scenarios that can occur when identifying or managing risks: (1) No risk is identified/managed, (2) Not all the risks are identified/managed, and (3) All the risks have been identified/managed. Considering these scenarios, we can adopt a three levels scale to evaluate the risk identification level, the risk management level and the SOLAP datacubes reliability level: the level is Null (N) in scenario (1), Medium (M) in scenario (2) and High (H) in scenario (3) in each case. The following matrix shows the possible scenarios of the risks identification and management levels impacts on the SOLAP datacube reliability level.

First, we note that a Null reliability is also what is observed for a classical SOLAP datacube where nothing is done to identify the risks of misinterpretation and/or manage them. Second, the only time a very high reliability level is reached is when 100% of the risks have been identified and 100% of those identified risks have been managed. It is the ideal scenario. When risks are missed during the identification step and only some of the identified ones are managed, as well as when all the risks have been identified but not all of them have been managed and finally when not all the risks have been identified but all of those identified are managed, the reliability level is Medium, which means it is still better than any classical SOLAP datacubes one.

		Risk identification level		
		0% risks identified (N)	Some risks not identified (M)	100% risks identified (H)
Risk management level	0% risks managed (N)	N	N	N
	Some risks not managed (M)	N	**M**	M
	100% risks managed (H)	N	**M**	H

Figure 6-1: Risks identification and management impact on reliability

In the event not all risks are identified, especially important ones (risk identification level at Null or Medium), end-users are still exposed to all the non-identified risks. They may even think that because the method was used to produce the given SOLAP datacube, no mention of a risk equals no risk. The same thing goes for the Null and Medium risks management levels. The most logical solution is probably to add a meta-uncertainty variable that would hold the level of reliability of the designed SOLAP datacubes. The question related to this solution is how to compute the reliability level during the design, which translated

into how to compute the identification and management levels. For the management level, it will come down to a ratio between the number of risks identified and the number of risk managed. For the risk identification level, we think that if vague geometric attributes are introduced in the datacube, at least the corresponding risk-geometrys should be identified. If it is not the case, and risks have been identified, we can definitely say that some risks are missing (level = Medium). If no risk has been identified at all, the level of identification and thus reliability is clearly Null. Automating identification process is probably the best way to prevent missing risks. It will also allow knowing for sure if the list of risks identified is exhaustive and therefore knowing if we are at 100% of risks identified. Two ways to do so are to base the identification process on a mapping file between vagueness and risks of misinterpretation created for each project or to exploit a risk ontology. For now, we can conclude that the method definitely allows designing SOLAP datacubes with a known Medium reliability level at most (Cf. bold M in Figure 6-1). It might be high in some cases but the meta-uncertainty parameter will not be able to express that since it is not certain.

6.3 Research perspectives

This thesis provided fundamentals, from a theoretical risk-aware approach all the way to a practical design method implementing that approach, to support a new and efficient manner to handle spatial vagueness in SOLAP datacubes. Doing so, and now that we know our approach is promising, the thesis opens the door for new research perspectives to enhance and validate the risk-aware approach as well as enhance SOLAP datacubes quality.

- **Enhance risk-aware approach:**
    - Vagueness representation: It could be interesting to introduce into the SOLAP datacubes, a grid representation of the vagueness that is not as complex as the fuzzy logic. This representation could be based on a rough set model or a new model exploiting the existing exact models. For instance, we think about a model where the spatial data is composed of 3 sets of points (minimum extent, vague extent and maximum extent) associated with values indicating the meta-uncertainty on their localization. Such a representation would help fine-tune the risks of misinterpretation identification and thus their reduction.
    - Risk ontology and/or knowledge base: a research perspective could be to go beyond our risk classification by proposing a SOLAP datacubes risk of misuse ontology. Such an ontology/knowledge base would define different type of risks and their sources as well as all the possible risk control actions that have been used before or that are preconized. It would allow a better risks identification, a better risk control and a better risk monitoring.
    - Risk identification automation: We believe that automating the risk identification process would help leverage the full power of our risk-aware approach. Indeed, it will guarantee the completeness of the risks identified throughout the design iterations and in consequence a better risk reduction. To do so, one idea would be to base the risk identification process on a

risk ontology or a risk file that should be defined at the beginning of each project where all potential risks have been listed knowing the spatial vagueness present on the sources.

- Risk communication: In this thesis, we didn't really tackle the risk visual communication on the client. It could be interesting to look deeper into the graphical semiology regarding the risks communication on the map as well as communication policies for diagrams and pivot table. The result could be for instance the proposition of a graphic representation for the risk-geometry and risk-aggregation (dashed boundaries, dotted interior etc.).

- **Validating the approach on more data, with different geometries and the participation of decision-makers**
  - This thesis concentrated on the validation of the approach from a theoretical point view. It would be interesting to test the approach on more case studies in real project contexts.
  - Also it would be interesting to define and validate usability criteria that are specific to this risk-aware SOLAP datacubes schemas, especially ones that will allow the testing of the usability of a combination of datacubes.
  - Finally, it would also be interesting to conduct a front-end testing with decision makers on the SOLAP datacube usability. The testing would be on the exploitation of both map and pivot tables.

- **Adapt the approach for other uncertainty or quality issues:** This thesis focused on what happens in SOLAP datacubes when spatial vague objects are involved. We believe that for other type of uncertainty or data quality issue, the risk-aware approach can be adopted to limit the errors related to the exploitation of geospatial databases in general. For instance, it could be interesting to adapt the risk-aware approach to deal with incompleteness in data sources. With such an approach, end-users will be provided with databases that might have missing data but still fit their needs. The risks of misuse would be taken care of according to their tolerance levels, in those databases. In the same vein it would be interesting to adapt the approach to temporal vagueness.

- **Adapt and integrate the approach in other design methods:** In this thesis, we have integrated the risk-aware design approach in a rapid prototyping method. However, it would be interesting to test the approach in others design methods or approaches. In particular, we think about the collaborative spatial database design framework proposed by Grira (2014) in his thesis. The result would be for example a collaborative framework where end-users are requested not only in the data usages description and risks identification as the approach originally advocates, but also in the risks assessment and control as advocated by our approach. Of course, the framework should be adapted to SOLAP datacubes design in particular because our contributions are limited to those type of databases.

References

Adamson, C. (2006). Mastering Data Warehouse Aggregates: Solutions for Star Schema Performance, John Wiley & Sons.

Beard, K. (1989). Use error: the neglected error component. Proceedings of Auto-Carto.

Bédard, Y. (1986). "A study of Data using a Communication based Conceptual Framework of land Information Systems." Le Géomètre canadien **40**(4): 12.

Bédard, Y. (1995). La qualité des données en géomatique. Géomatique V.

Bédard, Y. and J. Han (2009). Fundamentals of Spatial Data Warehousing for Geographic Knowledge Discovery. Geographic Data Mining and Knowledge Discovery. H. J. Miller and J. Han, Taylor & Francis.

Bédard, Y., S. Larrivée, M. Proulx, P. Caron and F. Létourneau (1997). "Étude de l'état actuel et des besoins de R&D relativement aux architectures et technologies des data warehouses appliquées aux données spatiales." Research report for the Canadian Defense Research Center in Valcartier.

Bédard, Y., T. Merrett and J. Han (2001). "Fundamentals of spatial data warehousing for geographic knowledge discovery." Geographic data mining and knowledge discovery **2**.

Bédard, Y., M. J. Proulx and S. Rivest (2005). "Enrichissement du OLAP pour l'analyse géographique: exemples de réalisation et différentes possibilités technologiques." Entrepôts de Données et Analyse en ligne, RNTI B_1. Paris: Cépaduès: 1-20.

Bédard, Y., S. Rivest and M.-j. Proulx (2006). Spatial On-Line Analytical Processing (SOLAP): Concepts, Architectures, and Solutions from a Geomatics Engineering Perspective. Data Warehouses and OLAP: Concepts, Architecture, and Solutions, Idea Group Publishing.

Bejaoui, L. (2009). Qualitative topological relationships for objects with possibly vague shapes: implications on the specification of topological integrity constraints in transactional spatial databases and in spatial data warehouses, Université Blaise Pascal.

Bejaoui, L., F. Pinet, Y. Bedard and M. Schneider (2009). "Qualified topological relations between spatial objects with possible vague shape." International Journal of Geographical Information Science **23**(7): 877-921.

Berenguer, G., R. Romero, J. Trujillo, M. Serrano and M. Piattini (2005). A Set of Quality Indicators and Their Corresponding Metrics for Conceptual Models of Data Warehouses. Data Warehousing and Knowledge Discovery. A. Tjoa and J. Trujillo, Springer Berlin Heidelberg. **3589:** 95-104.

Bimonte, S., H. Nazih, M.-A. Kang, E. Edoh-Alove and S. Rizzi (2013). ProtOLAP: Rapid OLAP Prototyping with On-Demand Data Supply. DOLAP'13. San Fransisco, CA, USA.

Bimonte, S., P. Wehrle, A. Tchounikine and M. Miquel (2006). GeWOlap: A Web Based Spatial OLAP Proposal. On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops. R. Meersman, Z. Tari and P. Herrero, Springer Berlin / Heidelberg. **4278:** 1596-1605.

Boehm, B. W. (1991). "Software risk management: principles and practices." Software, IEEE **8**(1): 32-41.

Böhnlein, M., M. Plaha and A. Ulbrich-vom Ende (2002). "Visual Specification of Multidimensional Queries based on a Semantic Data Model." Vom Data Warehouse zum Corporate Knowledge Center (DW): 379-397.

Bonifati, A., F. Cattaneo, S. Ceri, A. Fuggetta and S. Paraboschi (2001). "Designing data marts for data warehouses." ACM transactions on software engineering and methodology **10**(4): 452-483.

Boulil, K. (2012). Une approche automatisée basée sur des contraintes d'intégrité définies en UML et OCL pour la vérification de la cohérence logique dans les systèmes SOLAP: Applications dans le domaine agri-environmental. Doctorat, Université Blaise Pascal.

Boulil, K., S. Bimonte and F. Pinet (2011). "Un modèle UML et des contraintes OCL pour les entrepôts de données spatiales. De la représentation conceptuelle à l'implémentation." Ingénierie des Systèmes d'Information **16**(6): 11-39.

Boulil, K., S. Bimonte and F. Pinet (2012). A UML & Spatial OCL based approach for handling quality issues in SOLAP systems. Proceedings of the14th International Conference on Enterprise Information Systems (ICEIS 2012). Wroclaw, Poland**:** 6.

Boulil, K., S. Bimonte, F. Pinet and J.-P. Chanet (2012). "Conceptual Model for Spatial Data Cubes - A UML Profile and its Automatic Implementation." Computer Standards & Interfaces (submitted on 2012-08-29): 40.

Boutry, R., F. Gassion, B. Gallais, J. Molenaar, R. Bariller, M. Mathieu, M. Dimet, Orchestre de la Garde Républicaine., Franske Hærs Kor., Batterie-Fanfare de la Garde Républicaine., Maréchal des Logis-Chef Michel Moisseron., Fanfare de la Cavalerie de la Garde. and Ch@0153ur d'enfants des classes musicales d'Aulnay-sous-Bois. Révolution française, Garde Républicaine, EMI.

Burrough, P. A. and A. U. Frank (1996). Geographic Objects with Indeterminate Boundaries
London, Taylor & Francis.

Caron, P. Y. (1998). Etude du potentiel de OLAP pour supporter l'analyse spatio-temporelle. Master, Université Laval.

Chaudhuri, S. and U. Dayal (1997). "An overview of data warehousing and OLAP technology." SIGMOD Rec. **26**(1): 65-74.

Clementini, E., P. D. Felice and P. v. Oosterom (1993). A Small Set of Formal Topological Relationships Suitable for End-User Interaction. Proceedings of the Third International Symposium on Advances in Spatial Databases, Springer-Verlag**:** 277-295.

Codd, E. F., S. B. Codd, C. T. Salley, Codd and I. Date (1993). Providing OLAP (On-line Analytical Processing) to User-analysts: An IT Mandate, Codd amp; Associates.

Cohn, A. G. and N. M. Gotts (1996). "The 'egg-yolk'representation of regions with indeterminate boundaries." Geographic objects with indeterminate boundaries **2**: 171-187.

Del Cano, A. and M. P. de la Cruz (2002). "Integrated methodology for project risk management." Journal of Construction Engineering and Management **128**(6): 473-485.

Delgado, M., C. Molina, D. Sánchez, L. R. Ariza and M. A. Vila (2004). A flexible approach to the multidimensional model: The fuzzy datacube. Current Topics in Artificial Intelligence, Springer**:** 26-36.

Devillers, R., Y. Bédard and R. Jeansoulin (2005). "Multidimensional Management of Geospatial Data Quality Information for its Dynamic Use Within Geographical Information Systems." American Society for Photogrammetry and Remote Sensing (PE&RS) **71**(No. 2): 205-215.

Devillers, R., Y. Bedard, R. Jeansoulin and B. Moulin (2007). "Towards spatial data quality information analysis tools for experts assessing the fitness for use of spatial data." International Journal of Geographical Information Science **21**(3): 261-282.

Devillers, R. and R. Jeansoulin (2006). Fundamentals of spatial data quality. London, ISTE.

Devillers, R., A. Stein, Y. Bédard, N. Chrisman, P. Fisher and W. Shi (2010). "Thirty Years of Research on Spatial Data Quality: Achievements, Failures, and Opportunities." Transactions in GIS **14**(4): 387-400.

Di Tria, F., E. Lefons and F. Tangorra (2012). "Hybrid methodology for data warehouse conceptual design by UML schemas." Information and Software Technology **54**(4): 360-379.

Dilo, A. (2006). Representation of and reasoning with vagueness in spatial information : a system for handling vague objects. Met lit. opg. - Met samenvatting in het Engels, s.n.].

Duboisset, M., F. Pinet, M. A. Kang and M. Schneider (2005). "Precise modeling and verification of topological integrity constraints in spatial databases: from an expressive power study to code generation principles." Lecture Notes in Computer Science (ER) **3716**: 465-482.

Edoh-Alove, E., S. Bimonte, F. Pinet and Y. Bédard (2015). "New Design Approach to Handle Spatial Vagueness in Spatial OLAP Datacubes: Application to Agri-environmental Data." International Journal of Agricultural and Environmental Information Systems **6**(3).

Fasel, D. and D. Zumstein (2009). A fuzzy data warehouse approach for web analytics, Springer.

Fisher, P. F. (1999). "Models of Uncertainty in Spatial Data." Geographical Information Systems: Principles, Techniques, Management and Applications **1**: 15.

Gervais, M., Y. Bédard, R. Jeansoulin and B. Cervelle (2007). "Qualité des données géographiques : obligations juridiques potentielles et modèle du producteur raisonnable." Revue Internationale de Géomatique **17**(1): 33-62.

Gervais, M., Y. Bédard, S. Larrivée, S. Rivest and T. Roy (2012). Enquête canadienne sur la qualité des données géospatiales et la gestion du risque, Centre for Research in Geomatics, Laval University, Quebec City, Canada.

Gervais, M., Y. Bédard, M.-A. Levesque, E. Bernier and R. Devillers (2009). "Data Quality Issues and Geographic Knowledge Discovery." Geographic Data Mining and Knowledge Discovery: 99-115.

Giorgini, P., S. Rizzi and M. Garzetti (2005). Goal-oriented requirement analysis for data warehouse design. Proceedings of the 8th ACM international workshop on Data warehousing and OLAP, ACM.

Glorio, O. and J. Trujillo (2008). An MDA Approach for the Development of Spatial Data Warehouses. Data Warehousing and Knowledge Discovery. I.-Y. Song, J. Eder and T. Nguyen, Springer Berlin / Heidelberg. **5182:** 23-32.

Golfarelli, M. and S. Rizzi (2011). "Data warehouse testing." International Journal of Data Warehousing and Mining (IJDWM) **7**(2): 26-43.

Gosain, A., S. Nagpal and S. Sabharwal (2011). "Quality metrics for conceptual models for data warehouse focusing on dimension hierarchies." ACM SIGSOFT Software Engineering Notes **36**(4): 1-5.

Grira, J. (2014). Improving Kowledge About The Risks of Inappropriate Uses of Geospatial Data By Intoducing A Collaborative Approach In The Design Of Geospatial Databases. Ph.D., Laval University.

Grira, J., Y. Bédard and S. Roche (2013). Revisiting the Concept of Risk Analysis within the Context of Geospatial Database Design: A Collaborative Framework. World Academy of Science, Engineering and Technology 75, Madrid, Spain.

Guimond, L.-E. (2005). Conception d'un environnement de découvertes des besoins pour le développement de solutions SOLAP. Master, Université Laval.

Han, J., N. Stefanovic and K. Koperski (1998). Selective materialization: An efficient method for spatial data cube construction. Research and Development in Knowledge Discovery and Data Mining. X. Wu, R. Kotagiri and K. Korb, Springer Berlin / Heidelberg. **1394:** 144-158.

Hazarika, S. and A. Cohn (2001). Qualitative Spatio-Temporal Continuity Spatial Information Theory. D. Montello, Springer Berlin / Heidelberg. **2205:** 92-107.

ISO (2000). ISO 9000: Quality Management Systems:Fundamentals and Vocabulary.

ISO/IEC-51 (1999). ISO/IEC 51 Safety aspects - Guidelines for their inclusion in standards.

ISO/TC 211 (2002). ISO 19113:2002: Geographic information -- Quality principles

Jadidi, A., M. A. Mostafavi, Y. Bédard and B. Long (2012). Towards an Integrated Spatial Decision Support System to Improve Coastal Erosion Risk Assessment: Modeling and Representation of Risk Zones. FIG Working Week 2012. Rome, Italy**:** 6-10.

Jarke, M., M. Lenzerini, Y. Vassiliou and P. Vassiliadis (2003). Fundamentals of data warehouses, Springer-Verlag New York Inc.

Jensen, M., T. Holmgren and T. Pedersen (2004). "Discovering multidimensional structure in relational data." Data Warehousing and Knowledge Discovery: 138-148.

Karolak, D. W. and N. Karolak (1995). Software Engineering Risk Management: A Just-in-Time Approach, IEEE Computer Society Press.

Kerzner, H. (2006). Project Management: a systems approach to planning, scheduling, and controlling (9th Edition). New Jersey, United States.

Kimball, R. and M. Ross (2002). The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, 2nd Edition, Wiley.

Lagacherie, P., P. Andrieux and R. Bouzigues (1996). Fuzzyness and uncertainty of soil boundaries: from reality to coding in GIS. Geographic Objects with Indeterminate Boundaries, GISDATA series. P. A. Burrough and A. U. Frank. London, Taylor & Francis. **2:** 275-286.

Laurent, A. (2002). Bases de données multidimensionnelles floues et leur utilisation pour la fouille de données Ph. D. Thesis, Université Paris 6.

Lévesque, M.-A. (2008). Formal Approach for a better identification and management of risks of inappropriate use of geodecisional data. Master of Science, Laval University.

Lujan-Mora, S., J. Trujillo and I.-Y. Song (2006). "A UML profile for multidimensional modeling in data warehouses." Data & Knowledge Engineering **59**(3): 725-769.

Malinowski, E. (2014). GeoBI Architecture Based on Free Software: Experience and Reviewdsz. Geographical Information Systems: Trends and Technologies. E. Pourabbas, CRC Press**:** 244-286.

Malinowski, E. and E. Zimányi (2004). Representing spatiality in a conceptual multidimensional model. GIS**:** 12-22.

Malinowski, E. and E. Zimányi (2008). "Advanced Data Warehouse Design." Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications, Data-Centric Systems and Applications, Volume. ISBN 978-3-540-74404-7. Springer Berlin Heidelberg, 2008 **1**.

Malinowski, E. and E. Zimányi (2008). Designing Conventional Data Warehouses. Advanced Data Warehouse Design. Berlin, Springer - Verlag Berlin Heidelberg**:** 245-306.

Malinowski, E. and E. Zimányi (2008). Designing Spatial and Temporal Data Warehouses. Advanced Data Warehouse Design. Berlin, Springer - Verlag Berlin Heidelberg**:** 307-343.

Mazón, J.-N., J. Trujillo and J. Lechtenbörger (2007). "Reconciling requirement-driven data warehouses with data sources via multidimensional normal forms." Data Knowl. Eng. **63**(3): 725-751.

Naouali, S. and R. Missaoui (2005). Flexible query answering in data cubes. Data Warehousing and Knowledge Discovery, Springer**:** 221-232.

Naouali, S. and R. Missaoui (2006). "Approximation des cubes OLAP et génération de règles dans les entrepôts de données."

OMG (2003). MDA Guide, *Version 1.0.1*, Object Management Group.

OMG (2011). OMG Meta Object Facility (MOF) Core Specification - Version 2.4.1, Object Management Group.

OMG (2011). Unified Modeling Language™ (OMG UML), Infrastructure, *Version 2.4*, Object Management Group.

Pardillo, J., J.-N. Mazón and J. Trujillo (2010). "Designing OLAP schemata for data warehouses from conceptual models with MDA." Decision Support Systems **3**(1): 51-62.

Parent, C., S. Spaccapietra and E. Zimányi (2006). Conceptual modeling for traditional and spatio-temporal applications : The MADS approach. Berlin, Springer-Verlag.

Pauly, A. and M. Schneider (2010). "VASA: An algebra for vague spatial data in databases." Information Systems **35**(1): 111-138.

Pedersen, T. B. and N. Tryfona (2001). Pre-aggregation in Spatial Data Warehouses. Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases, Springer-Verlag**:** 460-480.

Perez, D., M. J. Somodevilla and I. H. Pineda (2007). Fuzzy Spatial Data Warehouse: A Multidimensional Model. Proceedings of the Eighth Mexican International Conference on Current Trends in Computer Science, IEEE Computer Society**:** 3-9.

Phipps, C. and K. C. Davis (2002). "Automating data warehouse conceptual schema design and evaluation." Lakshmanan [19]: 23-32.

Pitarch, Y., C. Favre, A. Laurent and P. Poncelet (2012). Enhancing flexibility and expressivity of contextual hierarchies. Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on, IEEE.

Prat, N., J. Akoka and I. Comyn-Wattiau (2006). "A UML-based data warehouse design method." Decis. Support Syst. **42**(3): 1449-1473.

Proulx, M. J. and Y. Bédard, Rivest, S., et M.-J. Proulx, (2004). Le potentiel de l'approche multidimensionnelle pour l'analyse de données géospatiales en comparaison avec l'approche transactionnelle des SIG. Colloque Géomatique 2004 - Un choix stratégique! Montréal, Canada.

Rafanelli, M. (2003). Multidimensional Databases: Problems and Solutions, IGI Global.

Rivest, S., Y. Bédard and P. Marchand (2001). "Towards better support for spatial decision-making: defining the characteristics of Spatial On-Line Analytical Processing." Geomatica **55**(4): 539-555.

Romero, O. and A. Abelló (2008). MDBE: Automatic Multidimensional Modeling. Conceptual Modeling - ER 2008. Q. Li, S. Spaccapietra, E. Yu and A. Olivé, Springer Berlin Heidelberg. **5231:** 534-535.

Romero, O. and A. Abelló (2009). A Survey of Multidimensional Modeling Methodologies, IGI Global.

Romero, O. and A. Abelló (2010). "Automatic validation of requirements to support multidimensional design." Data & Knowledge Engineering **69**(9): 917-942.

Romero, O. and A. Abelló (2011). Data-Driven Multidimensional Design for OLAP. Scientific and Statistical Database Management. J. Bayard Cushing, J. French and S. Bowers, Springer Berlin Heidelberg. **6809:** 594-595.

Roy, T. (2013). Nouvelle méthode pour mieux informer les utilisateurs de portails Web sur les usages inappropriés de données géospatiales. Master of Science, Laval University.

Salehi, M. (2009). Developing a Model and a Language to Identify and Specify the Integrity Constraints in Spatial Datacubes PhD, Faculté des études supérieures de l'Université Laval.

Salehi, M., Y. Bédard and S. Rivest (2010). " A Formal Conceptual Model and Definitional Framework for Spatial Datacubes." Geomatica **64**(3): 313-326.

Salka, C. (1998). Ending the ROLAP/MOLAP debate: usage based aggregation and flexible HOLAP. In Proceedings of 14th International Conference on Data Engineering.

Sampaio, M. C., A. G. d. Sousa and C. d. S. Baptista (2006). Towards a logical multidimensional model for spatial data warehousing and OLAP. Proceedings of the 9th ACM international workshop on Data warehousing and OLAP. Arlington, Virginia, USA, ACM**:** 83-90.

Sboui, T. (2010). A conceptual framework and a risk management approach for interoperability between geospatial datacubes. Ph. D., Université Laval.

Schneider, M. (1999). Uncertainty Management for Spatial Datain Databases: Fuzzy Spatial Data Types

Advances in Spatial Databases. R. Güting, D. Papadias and F. Lochovsky, Springer Berlin / Heidelberg. **1651:** 330-351.

Serrano, M., J. Trujillo, C. Calero and M. Piattini (2007). "Metrics for data warehouse conceptual models understandability." Information and Software Technology **49**(8): 851-870.

Servigne, S., N. Lesage and T. Libourel (2005). Composantes qualité et métadonnées. Qualité de l'Information géographique. R. Devillers and R. Jeansoulin. Paris, Hermès-Lavoisier**:** 205-237.

Siqueira, T. L. L., C. D. Aguiar Ciferri, V. C. Times and R. R. Ciferri (2012). Towards Vague Geographic Data Warehouses. Geographic Information Science. N. Xiao, M.-P. Kwan, M. Goodchild and S. Shekhar, Springer Berlin Heidelberg. **7478:** 173-186.

Siqueira, T. L. L., C. D. d. A. Ciferri, V. C. Times and R. R. Ciferri (2014). "Modeling vague spatial data warehouses using the VSCube conceptual model." GeoInformatica: 1-44.

Song, I.-Y., M. Piattini, Y.-P. P. Chen, S. Hartmann, F. Grandi, J. Trujillo, A. L. Opdahl, F. Ferri, P. Grifoni, M. C. Caschera, C. Rolland, C. Woo, C. Salinesi, E. Zimányi, C. Claramunt, F. Frasincar, G.-J. Houben and P. Thiran (2008). Advances in Conceptual Modeling - Challenges and Opportunities : ER 2008 Workshops CMLSA, ECDM, FP-UML, M2AS, RIGiM, SeCoGIS, WISM, Barcelona Spain, October 20-23, 2008. Proceedings. Berlin, Heidelberg, Springer Berlin Heidelberg.

Soulignac, V., F. Barnabé, D. Rat and F. David (2006). "SIGEMO: un système d'information pour la gestion des épandages de matières organiques - Du cahier de charges à l'outil opérationnel." Ingénieries(47): 37-42.

Stefanovic, N. (1997). Design and implementation of on-line analytical processing (OLAP) of spatial data PhD, Simon Fraser University.

Stefanovic, N., J. Han and 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.

Winter, R. and B. Strauch (2003). A method for demand-driven information requirements analysis in data warehousing projects. System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on, IEEE.

Worboys, M. (1998). "Computation with imprecise geospatial data." Computers, Environment and Urban Systems **22**(2): 85-106.

Zadeh, L. A. (1965). "Fuzzy sets." Information and Control **8**(3): 338-353.

Zimányi, E. and E. Malinowski (2008). Spatial Data Warehouses. Advanced Data Warehouse Design. Berlin Heidelberg, Springer-Verlag Berlin Heidelberg**:** 133-179.

Appendix A: Spatial vague objects typology

In our approach, we need to represent spatial objects with **single polygons** and not complex double to multiple polygons as in the QMM model. A single polygon can only render a part of a region with a broad boundary: either the minimal extent (or union of all minimal extents) or the maximal extent (or union of all maximal extents). Even though it is not the ultimate accurate model of the spatial vague object, having a polygon, representing a vague spatial object, be clearly identified as expressing a specific part of the phenomenon is already a good way to consider the spatial vagueness. It will also allow a highlighting of the spatial vagueness impacts on the datacube interpretation and reasoning around spatial vagueness reduction in general.

Based on this deduction, we propose to typify the spatial data before exploiting them. That way, a qualitative description of the spatial vagueness can be associated to each polygon, allowing data users to consider the spatial vagueness to some extent. This qualitative typology is based on the concept of region with a broad boundary. Indeed, the types identified are:

The *minimal extent* type(as defined in Bejaoui (2009) QMM Model): a minimal extent type can be associated with a polygon that represents a region where a vague phenomenon is certainly present.

The *maximal extent* type (as defined in Bejaoui (2009) QMM model): a maximal extent type can be associated with a polygon that represents a region where a vague phenomenon is probably present (including the minimal extent).

The *exact extent* type: this type is necessary to describe a crisp region or region with no broad boundary (minimal extent equals to the maximal extent in the Bejaoui (2009) QMM Model). Administrative departments can be represented for example by polygons with the type *exact extent* associated.

An example of vague spatial object is a flood area, the minimal extent being the riverbed and the maximal extent being the limits taken as far as possible (obtained for example by computing a buffer around the riverbed limits or by calculating the region at flood risk extent). A flood area should be represented with a single polygon in the context of a classic crisp representation. With our new typology, we have the choice to store:

- Only the riverbeds with the type *minimal extent* associated,
- Or the buffer limits with the type *maximal extent* associated.

In each case, the polygon characteristics (shape, perimeter, surface, position etc.) are different.

Depending on the type of spatial vague object exploited, in other words the part of the reality that is depicted, the data fitness for use and interpretation is impacted at different degrees.

In the SOLAP datacube, we can choose to consider the spatial vagueness (if applicable) only for geometric spatial levels of a geometric spatial dimension, for geometric spatial measures or for both. In this thesis, by considering the spatial vagueness, we mean introduce the new qualitative typology in the datacube. Whether the vagueness is considered only for levels or for measures or both, it is necessary in our approach to analyze the impacts of this uncertainty on the measures values and interpretation by end-users.

Our objective here is not to do an exhaustive, generic and complete analysis of all possibilities regarding the vagueness introduction in the SOLAP datacube; therefore we will first focus on the case where the vagueness is considered only for geometric spatial levels. What we need to do is to bring out some of the possible impacts to help define categories of risks of misinterpretation and develop our new risk-aware design approach later.

Before we begin, we hypothesize that all the members of a spatial level have the same geometry qualitative type (they are all minimal extents or maximal extents) and thus the level is homogeneously vague.

For the remaining of this chapter, we consider a simple case study were we have a geometric spatial hierarchy organized as follows:

- *Farming Plots* < *Flood Zones*

- A level *Farming Plots* grouping the farming plots; the regions with broad boundaries. The farming plots can thus be considered in their maximal extents (agricultural plots limits) or minimal extents (the limits of the part really cultivated) using single polygons.

- A level *Flood Zones*; as explained earlier, flood zones are regions with broad boundaries, the minimal extents being the riverbeds and the maximal extents being limits of a specific well calculated buffer.

We also have the quantity of pesticide spread in the farming plots (*QuantityPesticide*) and the surface of area spread with pesticide (*SurfaceSpread*) as measures.

Having only *maximal extents (MaxExtent)* or *minimal extents (MinExtent)* members in a geometric spatial level impacts the interpretation, and in some cases the measure values themselves, at that level.

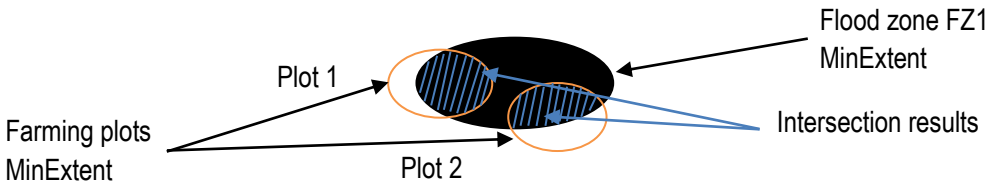
The measure, when linearly correlated to the shape or position of the member, will have values probably greater (or smaller) than what they should be because they are computing for the widest probable extent (or smallest extent) that represent the phenomenon. It is the case for example for measures relative to the surface of the member. For a flow (quantity/surface) on the member, it is the opposite because a division by the surface is made.

In our example, the values of the flow of pesticide will surely be over evaluated for the level *Farming Plots* if *minimal extents* are considered, or under evaluated if *maximal extents* are considered.

To conclude, depending on how the measure to be analyzed is defined, it is possible to deduce the cases of misinterpretation end-users are exposed to for a given level holding vague spatial objects.

In a hierarchy, it is possible to have different combinations of geometry types for two consecutive levels: MaxExtent < MaxExtent, MaxExtent < MinExtent, MaxExtent < ExactExtent, MinExtent < MaxExtent etc.

For each combination, the measure values aggregated for the parent members are impacted by the spatial vagueness on the child-members depending on the topological relationship between the two levels (Overlap, In, Disjoint etc.). The interpretation of those values can in turn be twice impacted, first by the aggregation itself, and second by the spatial vagueness on the parent members if applicable. For example, in the case of an overlap (the union of the child-members' geometries (farming plots limits) overlaps their parent's geometry (a polygon representing a flood zone)), one of the solutions to compute the measures for the parent would be to weight the measures according to the part of the child-members that are inside the parent geometry (Cf. Figure A-1 below).



	Total surface	Surface of the intersection with FZ1	SurfaceSpread for FZ1	QuantityPesticide for FZ1
Plot 1	1 ha	0,7 ha	0,7 ha	12600 kg
Plot 2	1 ha	0,5 ha	0,5 ha	6000 kg

Figure A-1: Illustration of the case where the topological relationship between Farming Plots and Flood Zones is Overlap

Appendix B: List of risks of misinterpretation associated with the Sludge SOLAP datacube

Measure	Aggregation level	Risks of poor measure evaluation	Example / Remarks
ProductFlowFromSludge	SpreadZones (MaxExtent)	Over-evaluation	4.09 g/m^2 instead of a value in [3.15, 4.09] for SpreadZone1 for example. We call this *Risk-Geometry.*
	Farms	Over-evaluation	The measure value for a given farm is the average of values for spread zones belonging to that farm. We call this *Risk-Aggregation.*
	Regions	Nothing to report	The flow value for a given region is the average of the values for all spread zones situated in that region. The great number of spread zones to take into account in the calculation allows the compensation of the uncertainties on farming plots flow values.
	Watersheds	Under-evaluation	The spread zone parts that pour into a given watershed are considered in the aggregation for that watershed (the value is weighted according to the farming plot surface that is included in the watershed surface). Because the spread zones are vague, there is an uncertainty about the intersections (with watersheds) resulting surfaces. It is also a risk *Risk-Aggregation*.
	Country	Nothing to report	Same reasoning as for the Regions level.

Table B-1: Risks of misinterpretation related to the measure ProductFlowFromSludge in the intended SOLAP datacube

Measure	Aggregation level	Risks of misinterpretation	Example / Remarks
ProductConcentrationInSoils	SpreadZones (MaxExtent)	Nothing to report	Because the tracking point is taken somewhere on green zones, the concentration value is well-evaluated when observed on green zones (which is the case in this intended datacube)
	Farms	Nothing to report	-
	Regions	Nothing to report	-
	Watersheds	Under-evaluation	The spread zone parts that pour into a given watershed are considered in the aggregation for that watershed. Because the spread zones are vague, there is an uncertainty about the intersections (with watersheds) resulting surfaces, thus on the aggregated measures. It is a *Risk-Aggregation.*
	Country	Nothing to report	-

Table B-2: Risks of misinterpretation related to the measure ProductConcentrationInSoils in the intended SOLAP datacube

Appendix C: RADSOLAP UML Profile metamodel

Here, we present the extended metamodels that describe the new risk-aware UML Profile. The profile is composed of the packages SDWCoreModelPackage and SDWAggregationModelPackage. The SDWCoreModelPackage comprises the SDWCoreRiskMetamodel (Cf. Figure C-1) for the datacube multidimensional structure definition and the SDWAttributeMetamodel (Cf. Figure C-2) for the attribute type definition. The SDWAggregationModelPackage is destined for the definition of the expected aggregations and thus contains the metamodel for describing the aggregation rules on the SDWAggregationRiskMetamodel (Cf. Figure C-3).

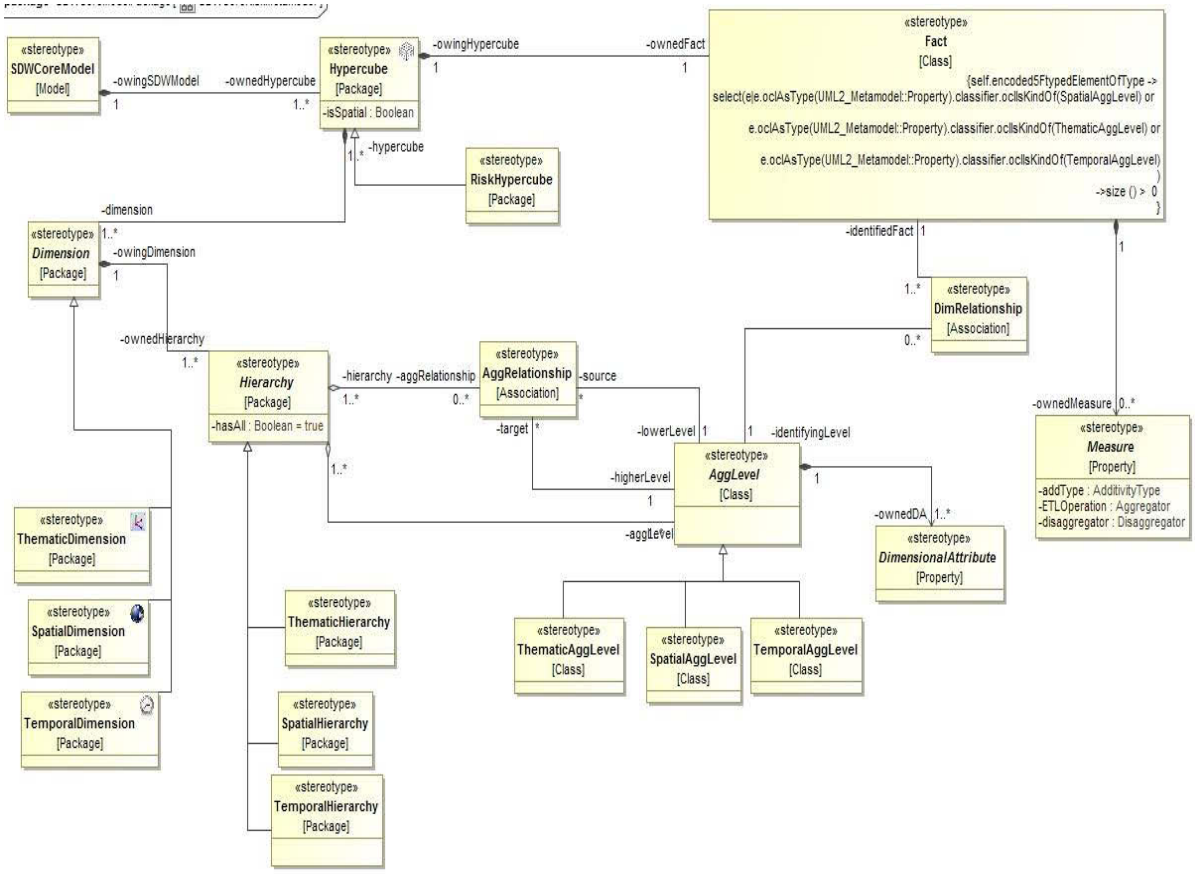


Figure C-1: Metamodel SDWCoreRiskMetamodel of the SDWCoreModelPackage

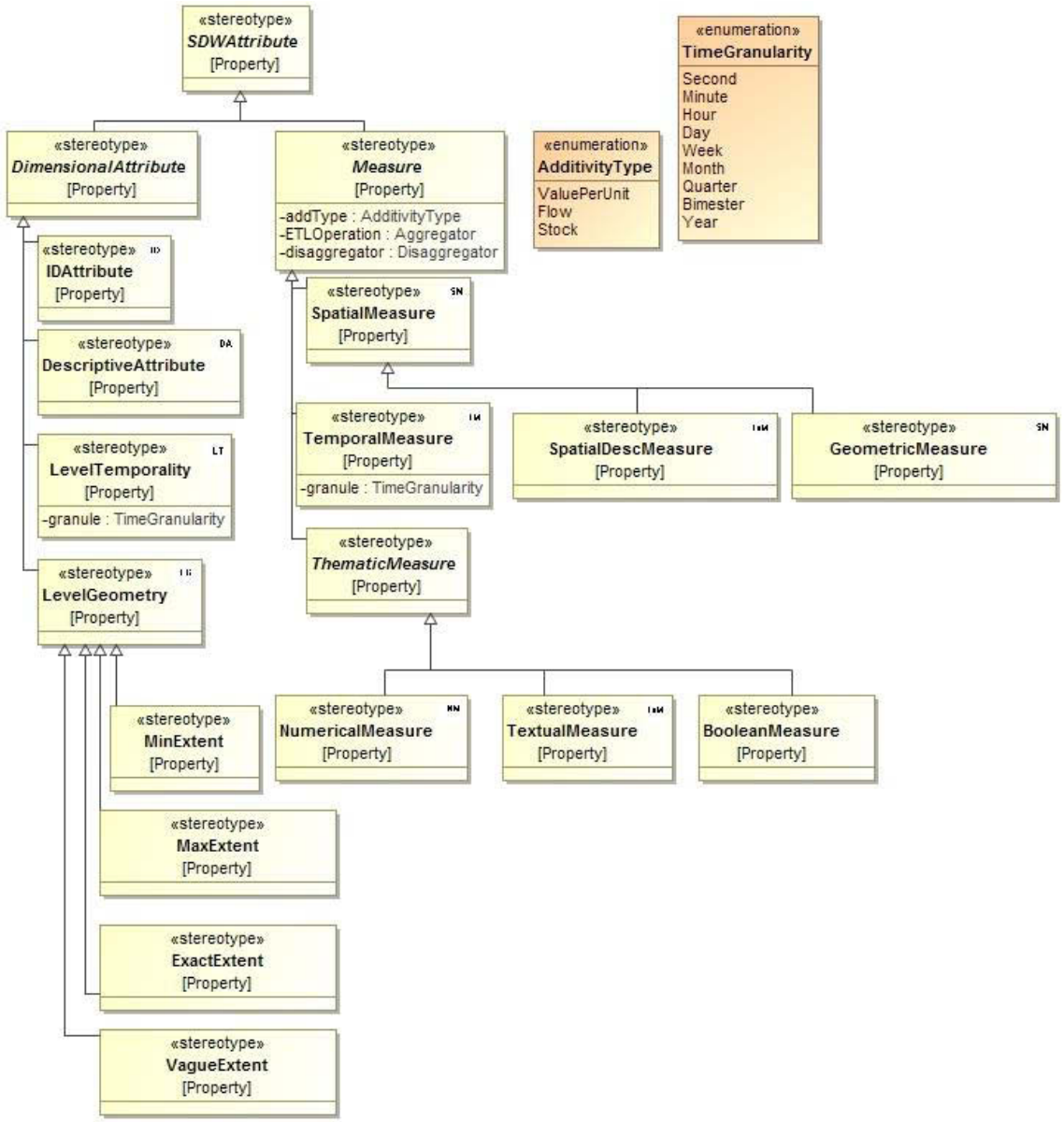


Figure C-2: Metamodel SDWAttributeMetamodel of SDWCoreModelPackage

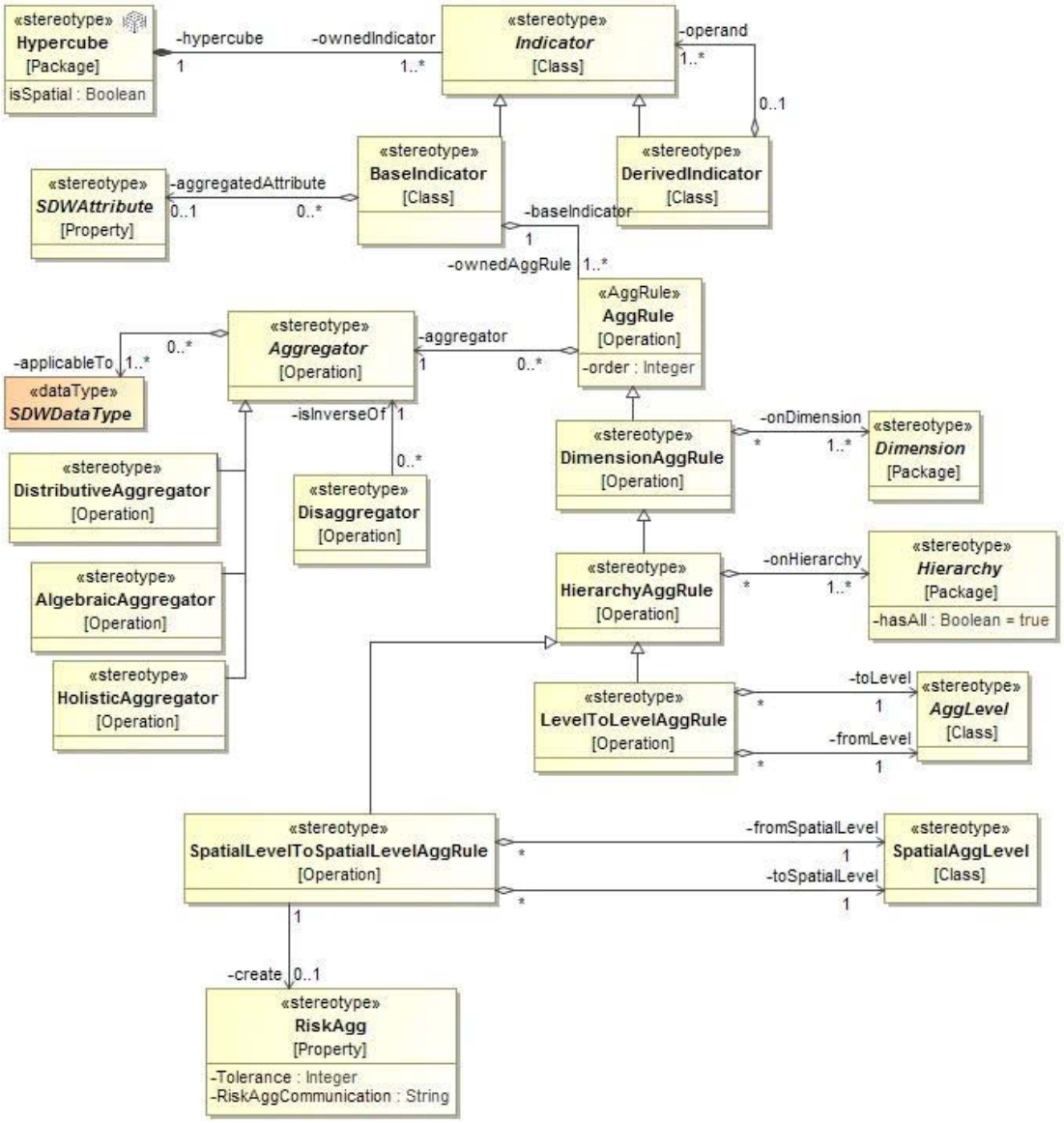


Figure C-3: Metamodel SDWAggregationRiskMetamodel of the SDWAggregationModelPackage

Appendix D: SOLAP datacube PIM transformation functions

In this section, we present the SOLAP datacube schemas transformation functions defined in the thesis. The complete list can be found in the following Table D-1.

DeleteSpatialLevel	**DeleteSpatialLevel (RiskHypercube Cr, SpatialLevel Cr.Ds.Hs.SpatialLevel[i]) = RiskHypercube Cr'. i ∈ {1, ..., n-1}, n being the number of levels in Hs.**
DeleteLowestSpatialLevel	DeleteLowestLevel (RiskHypercube Cr, SpatialDimension Cr.Ds) = RiskHypercube Cr'
DeleteHighestSpatialLevel	DeleteHighestLevel (RiskHypercube Cr, SpatialDimension Cr.Ds) = RiskHypercube Cr'
DeleteSpatialDimension	DeleteDimension (RiskHypercube Cr, SpatialDimension Cr.Ds) = RiskHypercube Cr'
ModifySpatialLevel	ModifySpatialLevel (RiskHypercube Cr, SpatialLevel Cr.Ds.Hs.SpatialLevel[i], String NewStereotype) = RiskHypercube Cr', i ∈ {0, n} , n being the number of spatial level
ModifyAggRuleAggregator	ModifyAggRuleAggregator (RiskHypercube Cr, AggRule Cr.BaseIndicator.AggRuleSL[i], Aggregator NewAggregator) = RiskHypercube Cr'.
ModifyAggRuleFromLevel	ModifyAggRuleFromLevel (RiskHypercube Cr, AggRule Cr.BaseIndicator.AggRuleSL[i], SpatialLevel Cr.Ds.Hs.SpatialLevel[j]) = RiskHypercube Cr', i ∈ {0, n}, and j ∈ {0, i-1}, n being the number of spatial level
ModifyRiskAgg	ModifyRiskAgg (RiskHypercube Cr, RiskAgg Cr.Ds.Hs.SpatialLevel[i].RiskLevel.Ragg[j], String newRiskAggValue) = RiskHypercube Cr', i ∈ {0, n}, n being the number of spatial level and j ∈ {1, m} m being the number of RiskAgg attributes in the RiskLevel class,
ModifyRiskGeom	ModifyRiskGeom (RiskHypercube Cr, RiskGeom Cr.Ds.Hs.SpatialLevel[i].RiskLevel.Rgeom, String newRiskGeomValue) = RiskHypercube Cr', i ∈ {0, n}, n being the number of spatial level
ModifyRiskGeomComm	ModifyRiskGeomComm (RiskHypercube Cr, RiskGeom Cr.Ds.Hs.SpatialLevel[i].RiskLevel.Rgeom, String policy) = RiskHypercube Cr', i ∈ {0, n}, n being the number of spatial level
ModifyRiskAggComm	ModifyRiskAggComm (RiskHypercube Cr, RiskAgg Cr.Ds.Hs.SpatialLevel[i].RiskLevel.Ragg[j], String policy) = RiskHypercube Cr', i ∈ {0, n}, n being the number of spatial level and j ∈ {1, m} m being the number of RiskAgg attributes in the RiskLevel class,

Table D-1: Formal definition of SOLAP datacubes schema transformation functions

DeleteSpatialLevel

This function deletes a spatial level in a spatial hierarchy. The level to be deleted is neither the highest nor the lowest level of the hierarchy. The function takes as input a RiskHypercube and the spatial level to be

deleted and as output we have a new RiskHypercube where the level in addition with its related aggregation relationships, RiskLevel class and aggregation rule are all deleted.

DeleteSpatialLevel (RiskHypercube *Cr*, SpatialLevel *Cr.Ds.Hs.SpatialLevel[i]*) = RiskHypercube *Cr'*.
i ∈ {1, ..., n-1}, n being the number of levels in Hs.

Such as,

Cr' = Cr

– {Cr.Ds.Hs.SpatialLevel[i], Cr.Ds.Hs.AggRel[i-1], Cr.Ds.Hs.AggRel[i], Cr.Ds.Hs.SpatialLevel[i].RiskLevel}

– Cr.BaseIndicator.AggRuleSL[i]

where Cr.BaseIndicator.AggRuleSL[i].fromSpatialLevel is SpatialLevel[i] or Cr.BaseIndicator.AggRuleSL[i].toSpatialLevel is SpatialLevel[i]

+ {new Cr.Ds.Hs.AggRel[i-1] with Cr.Ds.Hs.AggRel[i-1].lowerLevel = Cr.Ds.Hs.SpatialLevel[i-1] and Cr.Ds.Hs.AggRel[i-1].higherLevel = Cr.Ds.Hs.SpatialLevel[i+1]}

If Cr.Ds.Hs.SpatialLevel[i-1] and Cr.Ds.Hs.SpatialLevel[i+1] do not exist (a hierarchy with one level)

Cr' = Cr – {Cr.Ds.Hs.SpatialLevel[i].RiskLevel, Cr.Ds.Hs.DimRel, Cr.Ds}

– Cr.BaseIndicator. AggRuleSL[i], where Cr.BaseIndicator. AggRuleSL[i].fromSpatialLevel is SpatialLevel[i] or Cr.BaseIndicator. AggRuleSL[i].toSpatialLevel is SpatialLevel[i]

DeleteHighestSpatialLevel

This function deletes the highest level of the hierarchy. The function takes as input a RiskHypercube and the dimension from where the highest level should be deleted. The output is a new RiskHypercube where the level, the associated Risk Level, the related aggregation relationship and aggregation rule are all deleted.

DeleteHighestLevel (RiskHypercube *Cr,* SpatialDimension *Cr.Ds*) = RiskHypercube *Cr'*

Such as,

Cr' = Cr

– {Cr.Ds.Hs.SpatialLevel[n], Cr.Ds.Hs.AggRel[n-1], Cr.Ds.Hs.SpatialLevel[n].RiskLevel}

– Cr.BaseIndicator.AggRuleSL[n],

where Cr.BaseIndicator. AggRuleSL[n].fromSpatialLevel is SpatialLevel[n] or Cr.BaseIndicator. AggRuleSL

[n].toSpatialLevel is SpatialLevel[n]

DeleteSpatialDimension

This function deletes a spatial dimension in the hypercube. The function takes as input a RiskHypercube and the dimension to be deleted. The output is a new RiskHypercube where the dimension, the dimension relationship, the related risk package and the BaseIndicator are all deleted.

DeleteDimension (RiskHypercube *Cr,* SpatialDimension *Cr.Ds)* = RiskHypercube *Cr'* such as

Cr' = Cr – {Cr.Ds.Hs.DimRel, Cr.R, Cr.BaseIndicator}

ModifySpatialLevel

This function modifies the stereotype of the spatial level geometric attribute. The function takes as input a RiskHypercube, the spatial level to modify and the name (MinExtent, MaxExtent, ExactExtent) of new stereotype to apply.

ModifySpatialLevel (RiskHypercube *Cr*, SpatialLevel*Cr.Ds.Hs.SpatialLevel[i]*, String *NewStereotype*) = RiskHypercube *Cr'*, i∈ {0, n}, n being the number of spatial level such as

Cr' = Cr with Cr.Ds.Hs.SpatialLevel[i].level_geom.AppliedStereotype is *NewStereotype*

ModifyAggRuleAggregator

This function modifies the aggregator defined by an aggregation rule. The function takes as input a RiskHypercube, the aggregation rule to modify and the name of the new aggregator to apply and returns a new RiskHypercube where the aggregator value is set to the new one.

ModifyAggRuleAggregator (RiskHypercube *Cr*, AggRule*Cr.BaseIndicator.AggRuleSL[i]*, Aggregator*NewAggregator*) = RiskHypercube *Cr'*. i∈ {1…n}, n being the number of spatial levels such as

Cr' = Cr with Cr.BaseIndicator.AggRuleSL[i].aggregator = *NewAggregator*

ModifyAggRuleFromLevel

This function modifies the tag fromSpatialLevel value in the aggregation rule. The function takes as input a RiskHypercube, the aggregation rule to modify and the name of the spatial level to use for the

aggregation; and it returns a new RiskHypercube where the fromSpatialLevel value is set to the new given spatial level.

ModifyAggRuleFromLevel (RiskHypercube *Cr*, AggRule Cr.*BaseIndicator.AggRuleSL[i],* SpatialLevel *Cr.Ds.Hs.SpatialLevel[j]*) = RiskHypercube *Cr'*, i∈ {0, n}, and j ∈ {0, i-1}, n being the number of spatial level such as

Cr' = Cr with Cr.BaseIndicator.AggRuleSL[i].fromSpatialLevel = Cr.Ds.Hs.SpatialLevel[j]

ModifyRiskAgg

This function modifies the Risk-Aggregation attributes in the RiskLevel class. The function takes as input a RiskHypercube, the attribute RiskAgg to modify and the new value to affect to that attribute and returns a new RiskHypercube where the attribute RiskAgg is set to the new given value.

ModifyRiskAgg (**RiskHypercube** Cr, **RiskAgg** Cr.Ds.Hs.SpatialLevel[i].RiskLevel.Ragg[j], **String** newRiskAggValue**)** = **RiskHypercube** Cr', i∈ {0, n}, n being the number of spatial level and j ∈ {1, m} m being the number of RiskAgg attributes in the RiskLevel class,

Such as Cr' = Cr with Cr.Ds.Hs.SpatialLevel[i].RiskLevel.Ragg[j] = *newRiskAggValue*

ModifyRiskGeom

This function modifies the Risk-Geometry attributes in the RiskLevel class. The function takes as input a RiskHypercube, the attribute RiskGeom to modify and the new value to affect to that attribute and returns a new RiskHypercube where the attribute RiskGeom is set to the new given value.

ModifyRiskGeom (**RiskHypercube** Cr, **RiskGeom** Cr.Ds.Hs.SpatialLevel[i].RiskLevel.Rgeom, **String** newRiskGeomValue**)** = **RiskHypercube** Cr', i∈ {0, n}, n being the number of spatial level

Such as Cr' = Cr with Cr.Ds.Hs.SpatialLevel[i].RiskLevel.Rgeom = *newRiskGeomValue*

ModifyRiskGeomComm

This function modifies a Risk-Geometry communication tag. The function takes as input a RiskHypercube, the attribute RiskGeom for which the communication policy tag is to modify and the new policy and returns a RiskHypercube where the Risk-Geometry communication tag is set to the new policy.

ModifyRiskGeomComm (RiskHypercube *Cr*, RiskGeom *Cr.Ds.Hs.SpatialLevel[i].RiskLevel.Rgeom,* String *policy)* = RiskHypercube *Cr'*, i∈ {0, n}, n being the number of spatial level such as

Cr' = Cr with Cr.Ds.Hs.SpatialLevel[i].RiskLevel.Rgeom.RiskGeomCommunication = *policy*

policy domain values = {DefaultVisio, MapContourRed, MapContourGreen, CellColorRed, CellColorGreen, MakeAlert}.

ModifyRiskAggComm

This function modifies a Risk-Aggregation communication tag. The function takes as input a RiskHypercube, the attribute Risk-Aggregation for which the communication policy tag is to modify and the new policy and returns a RiskHypercube where the Risk-Aggregation communication tag is set to the new policy.

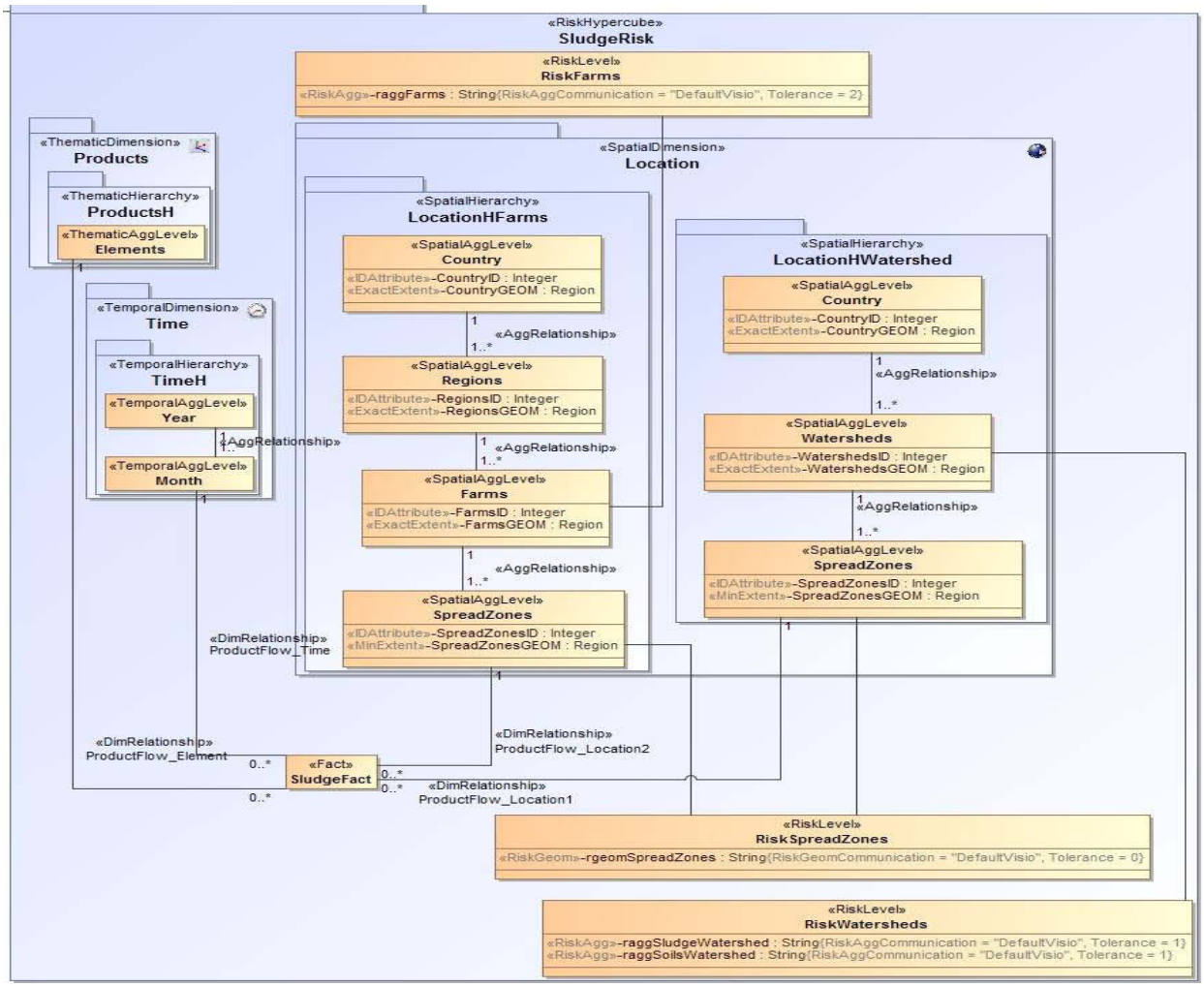
ModifyRiskAggComm (RiskHypercube Cr, RiskAgg Cr.Ds.Hs.SpatialLevel[i].RiskLevel.Ragg[j], String policy) = RiskHypercube Cr', $i \in \{0, n\}$, n being the number of spatial level and $j \in \{1, m\}$ m being the number of RiskAgg attributes in the RiskLevel class,

Such as

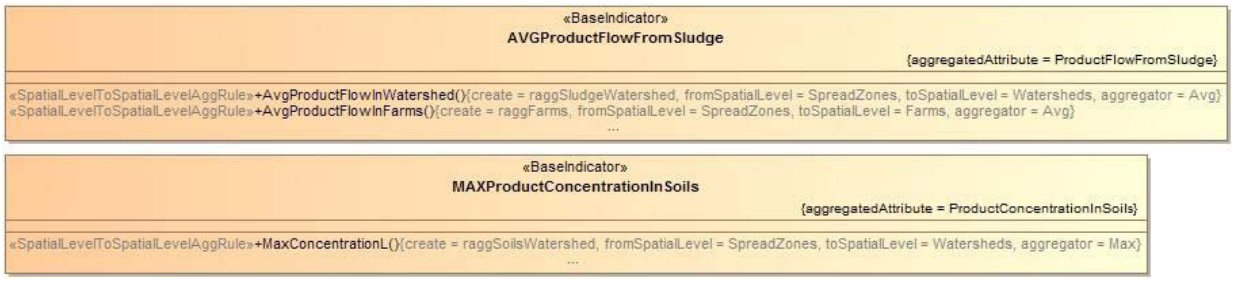
Cr' = Cr with Cr.Ds.Hs.SpatialLevel[i].RiskLevel.Ragg[j].RiskAggCommunication = *policy*

policy domain values = {DefaultVisio, CellColorRed, CellColorGreen, MakeAlert}

Appendix E: Initial PIM and elementary PIMs for the Sludge case study



a)



b)

Figure E-1: a) Initial intended SOLAP datacube multidimensional schema according to our RADSOLAP method; b) Initial BaseIndicators according to our RADSOLAP method

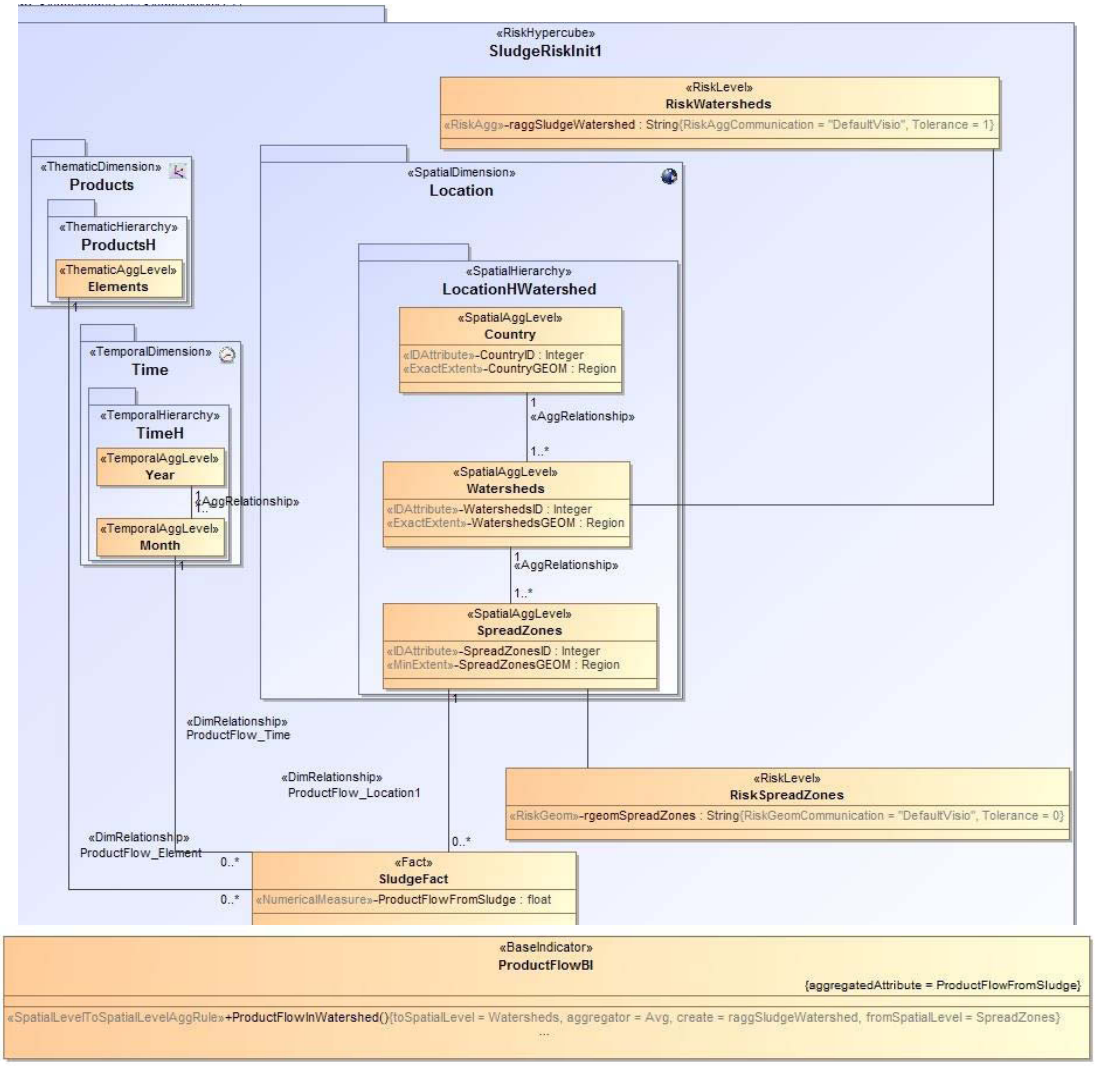


Figure E-2 : SludgeRiskInit1 PIM

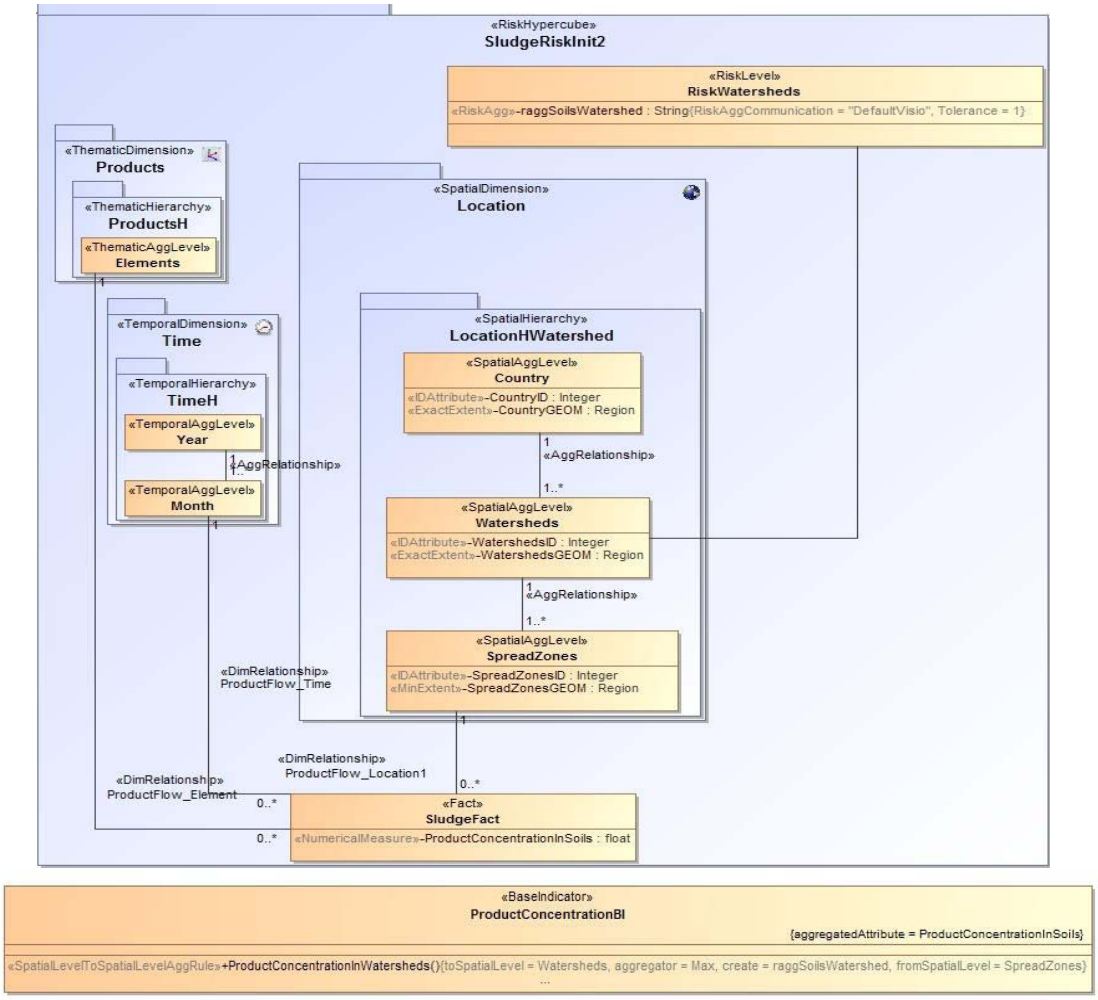


Figure E-3 : SludgeRiskInit2 PIM

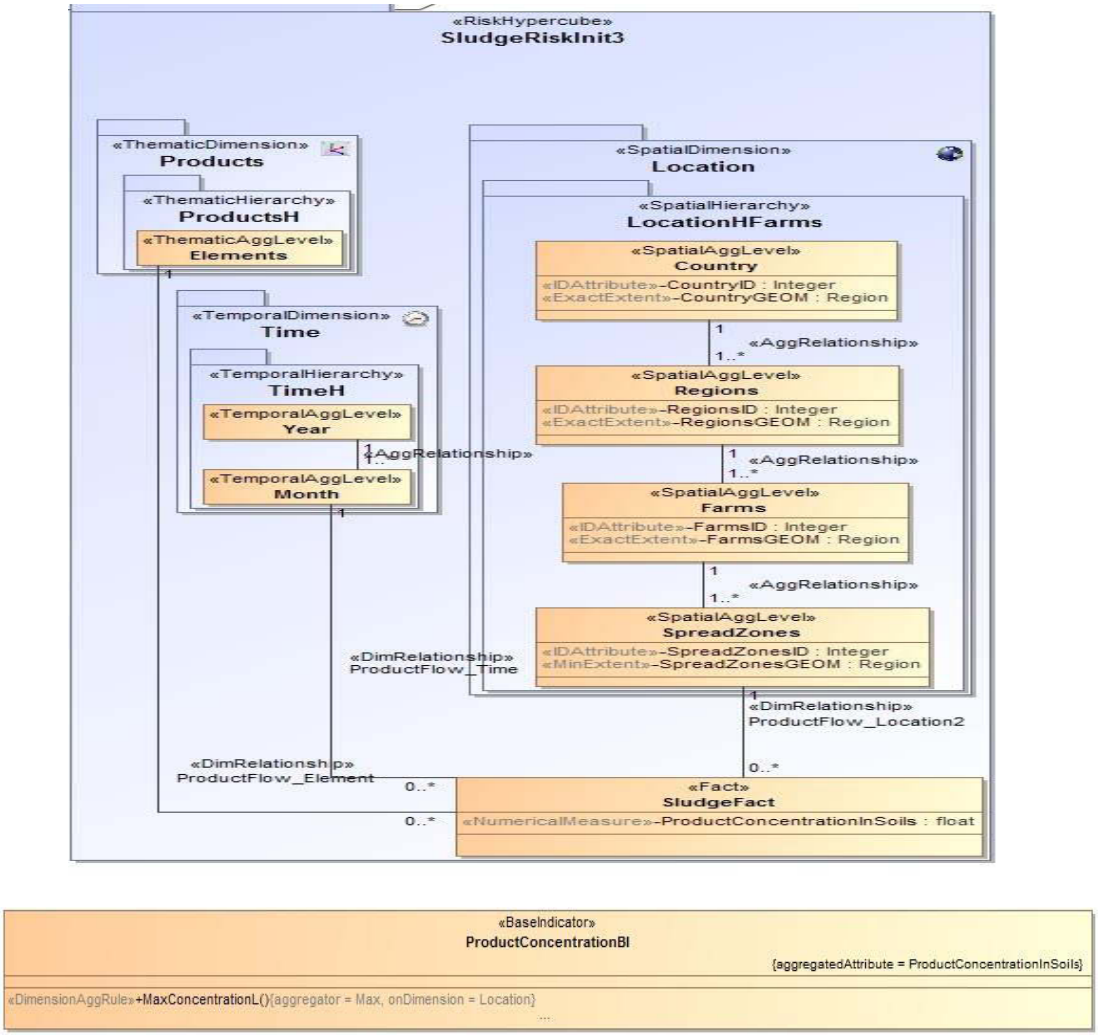


Figure E-4 : SludgeRiskInit3 PIM

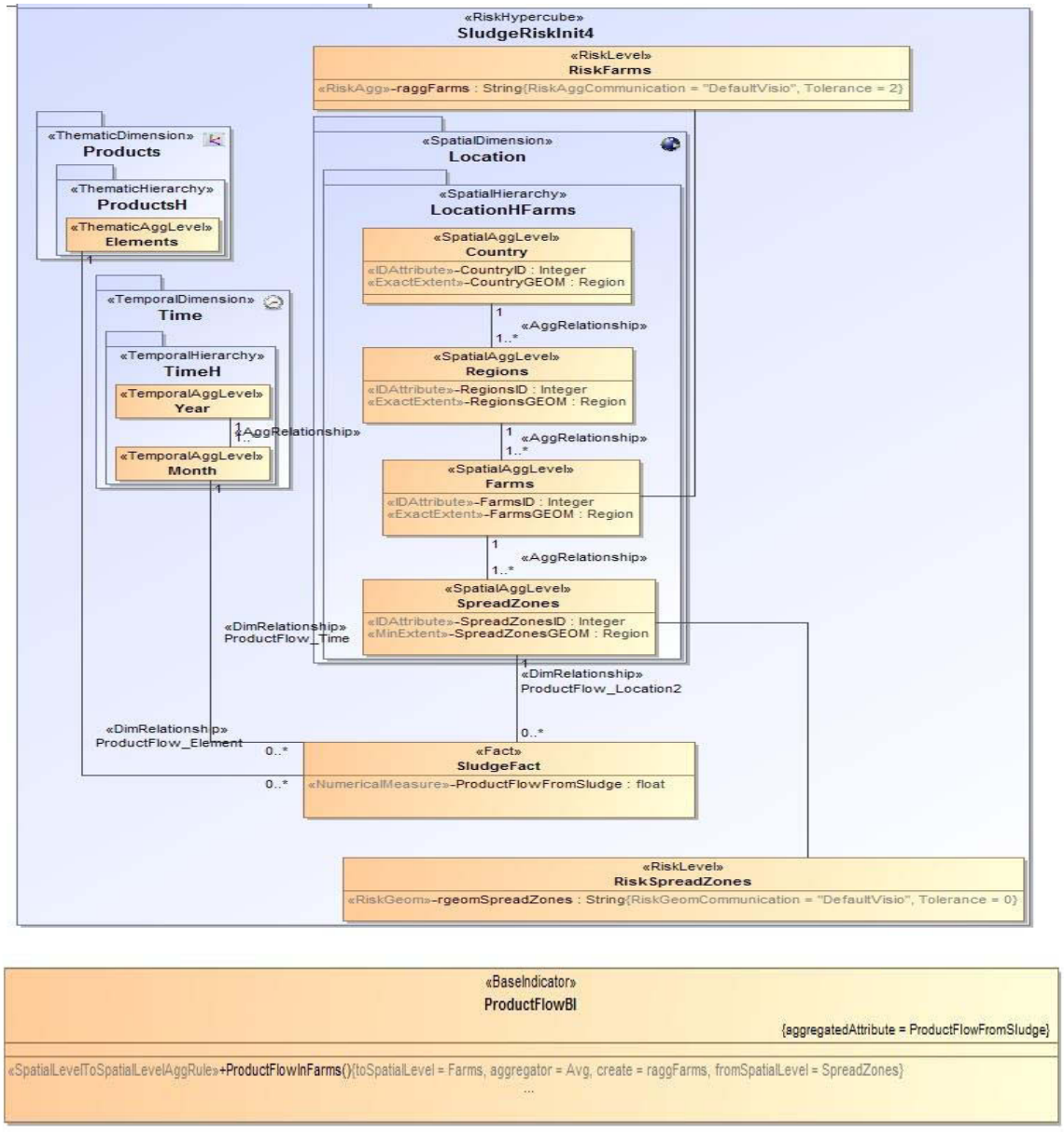


Figure E-5 : SludgeRiskInit4 PIM