Oper Res Int J (2010) 10:349–369
DOI 10.1007/s12351-009-0069-z

ORIGINAL PAPER

# Precise design of environmental data warehouses

**François Pinet · Michel Schneider**

Received: 22 December 2008 / Revised: 6 October 2009 / Accepted: 16 October 2009 /
Published online: 5 November 2009

**Abstract** People use data warehouses to help them make decisions. For example, public policy decision-makers can improve their decisions by using this technology to analyze the environmental effects of human activity. In production systems, data warehouses provide structures for extracting the knowledge required to optimize systems. Designing data warehouses is a complex task; designers need flexible and precise methods to help them create data warehouses and adapt their analysis criteria to developments in the decision-making process. In this paper, we introduce a flexible method based on UML (Unified Modeling Language). We introduce a UML profile for building multi-dimensional models and for choosing different criteria according to analysis requirements. This profile makes it possible to specify integrity constraints in OCL (Object Constraint Language). We apply our method to the construction of an environmental system for analyzing the use of certain agricultural fertilizers. We integrate various data sources into a multi-dimensional model showing several categories of analysis, and the consistency of data can be checked with OCL constraints.

**Keywords** Data warehouse · Environmental data · Agriculture ·
Decision support system · Unified Modeling Language ·
Object Constraint Language

## 1 Introduction

Along with the development of new information and communication technologies, we have seen a huge increase in sources of geo-referenced agricultural and

F. Pinet (✉) · M. Schneider
Cemagref, 24 Avenue des Landais, 63172 Aubière Cedex, France
e-mail: francois.pinet@clermont.cemagref.fr

M. Schneider
Blaise Pascal University, LIMOS, Clermont Ferrand, France

environmental data. Sensors and remote sensing systems acquire certain data. For example, systems use networks of sensors and satellite images to monitor work being done in agriculture. People enter other data with specialized computer programs (for example, using a computer application that records agricultural practices).

This information requires effective storage methods, as well as effective integration and analysis methods. Data warehouses are a specific type of database that serves to integrate, accumulate and analyze data from various sources (Calì et al. 2003). Users can load information from different databases into a data warehouse for combined analysis. Depending on their requirements, users can load data every week, every month, every year or even less frequently. These data are usually organized in a form that speeds up calculation of indicators. The indicators are made up of aggregated information obtained by functions such as sum, average, variance, etc. The main results of calculations are usually stored directly in the data warehouse, enabling rapid access.

Here we present an example from Trujillo et al. (2001). The *facts* of a data warehouse are the data to analyze (Malinowski and Zimanyi 2008). In the example, we consider the facts of the data warehouse to be the product sales of a company. Each of the company's stores provides these data. In a data warehouse, an analysis results from the use of an aggregation operation (e.g., sum or average) on the facts. In the example, a possible analysis is the sum of sales calculated by category of product, by store and by month. The result of this analysis can be represented in a cube (Trujillo et al. 2001)—see Fig. 1a. Each dimension of the cube corresponds to a criterion of analysis: type of products, store and month. The cells of the cube are called *measures*. They store the sums of sales for each tuple <type of products, store, month>. For instance, in Fig. 1a, the sum of sales for the tuple <Water, Store 1, December> is 4. In data warehouses, the criteria of analysis are structured in hierarchies called *dimensions*. Figure 1b shows the three dimensions presented by Trujillo et al. (2001). A data warehouse can produce many analyses by combining different levels of dimensions. For example, other cubes could be calculated:

- sums of sales by city,
- sums of sales by brand, by city, by year,
- sums of sales by type, by state, by season, etc.

Note that data warehouses generally support *n*-dimensional cubes. Data can be combined to provide previously unknown causal links. To do so, users can visualize cubes from the data warehouse using tools like OLAP (On-line Analytical Processing) (Malinowski and Zimanyi 2006, 2008). Causal links can also be discerned automatically with data-mining algorithms (Berson and Smith 1997).

Using data warehouses is therefore important within a decision-making context. For example, a data warehouse containing economic, urban and environmental information will help decision-makers find the best place to establish a new infrastructure. The concept of the data warehouse has great potential for assessing the impact of actions, practices, scenarios and programs from both the socio-economic and the environmental point of view (Schneider 2008). Two examples of use in agriculture can be found in (Nilakanta et al. 2008; Schulze et al. 2007).

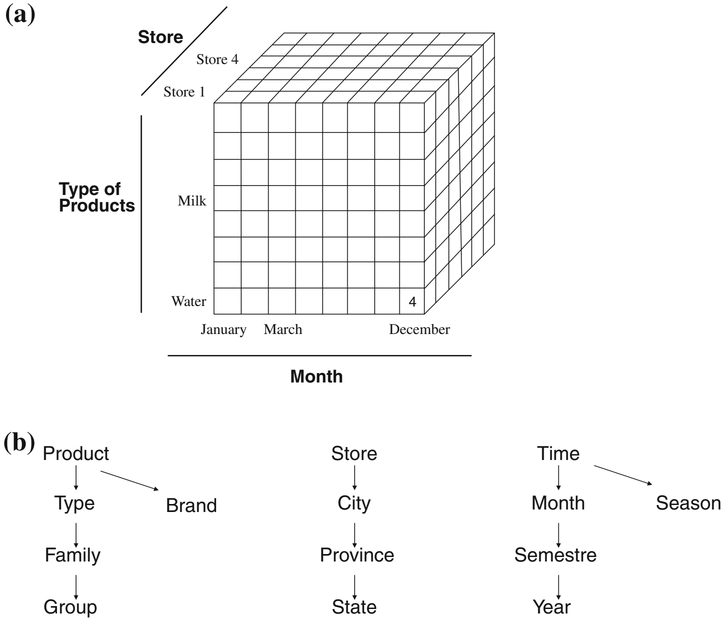

**Fig. 1** Example of a data warehouse. **a** Cube storing sales by category of products, by store and by month, **b** Analysis dimensions

Conceptual multi-dimensional models aim to describe the facts and the different analysis dimensions of a data warehouse (Malinowski and Zimanyi 2008). Recently, some articles have presented specific methods for formalizing multi-dimensional models (Lujan-Mora et al. 2006; Malinowski and Zimanyi 2008; Prat et al. 2006). Although existing design methods are efficient, there is room for further improvement.

Existing methods suppose, among other things, that the facts and dimensions are fixed (Lujan-Mora et al. 2006; Malinowski and Zimanyi 2008; Prat et al. 2006). However, for the same data, it may be necessary to apply different categories of analyses; for example, to calculate in the same data warehouse the sums of sales and the average prices of products. To mitigate this problem, in this paper we give a formalism to describe several categories of analysis within a single model. We represent multi-dimensional models by UML class diagrams. In our formalism, a class can represent both facts and the level of a dimension. This makes it possible to describe different categories of analysis focusing on the same data.

While methods for representing and checking integrity constraints in transactional databases already exist (Demuth and Hußmann 1999; Demuth et al. 2001), no such work has yet been done for data warehouses. However, if data are not integrated correctly, inconsistencies can appear in the warehouses. To address this problem, we present a method based on Object Constraint Language (OCL) (Schmid et al. 2002) that makes it possible to formalize rules that data warehouses need to respect. Checking these rules will ensure that analyses are not distorted by inconsistencies and errors in the data.

Our formalism for flexible multi-dimensional models is based on a new UML profile. Figure 2 describes certain stages of data warehouse design, with consideration given, among other things, to our UML profile, OCL language, data-mining and optimization algorithms. Of course, iterations between stages are possible.

Multi-dimensional models are formalized with UML and OCL. Case Tools could generate data warehouse physical schemata (Malinowski and Zimanyi 2008) (step 1). Several data sources can be integrated into data warehouses (step 2). Specific tools can facilitate the data integration. For example, ETL tools are software packages that can be used to Extract, Transform, and Load data from different sources into a data warehouse (Trujillo and Luján-Mora 2003). Case Tools could produce code to check integrity constraints in the integrated data (step 3). As presented in this paper, integrity constraints can be formalized in OCL. There are many tools for generating data verification mechanisms using OCL constraints (Demuth and Hußmann 1999; Demuth et al. 2001, 2004, 2005; Pinet et al. 2005, 2007, 2009; Klasse Objecten 2008). Causal links can also be discerned automatically by data-mining algorithms (step 4). Class diagrams are important to model data warehouses but other UML diagrams (such as sequence and use case) could be also considered at certain steps (e.g. integration).

Generally speaking, new techniques deriving from software engineering (such as UML and OCL) have been able to provide significant help in more precisely modeling computing systems in the agricultural and environmental fields (Muzy et al. 2005; Papajorgji 2007; Papajorgji and Shatar 2004; Papajorgji et al. 2004; Papajorgji and Pardalos 2006; DEAS 2008). Our work on UML and OCL modeling of environmental data warehouses follows this trend.

Section 2 introduces the example that will illustrate our ideas. The example is an application in the field of agricultural spreading. Section 3 describes our method for formalizing flexible multi-dimensional models based on a UML profile; this profile is important for the first step presented in Fig. 2. Section 4 exemplifies the data integration phase (step 2—Fig. 2). We then explain how to use OCL for describing data warehouse integrity constraints (step 3). Section 5 lists the principal methods

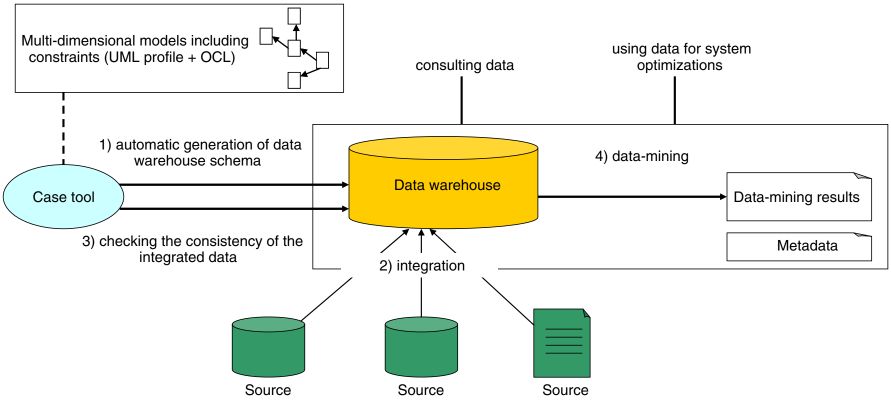
Multi-dimensional models including constraints (UML profile + OCL)
consulting data
using data for system optimizations
1) automatic generation of data warehouse schema
Case tool
Data warehouse
4) data-mining
Data-mining results
Metadata
3) checking the consistency of the integrated data
2) integration
Source
Source
Source

**Fig. 2** Using our UML profile and OCL in data warehouse design

for operating the data warehouses and provides examples within the context of data-mining and system optimization (step 4). Section 6 presents our conclusions and future work.

## 2 A System for the traceability of spreading in France

In this section, we present a case study inspired by the conceptual model of an information system for the traceability of agricultural spreading in France (Soulignac et al. 2005; Pinet et al. 2009). This operational system, called Sigemo, is used throughout France and was devised at Cemagref (2008). Currently, Sigemo includes a traditional transactional database. In this section, we show how some data from Sigemo could be used in a data warehouse. We show how the conceptual model of the current transactional database could be exploited in order to build the multi-dimensional model of a data warehouse. This example will illustrate our ideas throughout the paper.

Currently, the system records various plans for spreading organic matter on agricultural land in France. A spreading project consists of estimates for spreading a series of matter over a given period. Information is stored in a national transactional database. Research consultancies are responsible for devising the spreading projects. They work on behalf of producers of organic matter, which are mainly water treatment plants, livestock installations and the food industries. Using a specific web application, the research consultancies enter details about the spreading estimates in the database. The model in Fig. 3 shows the part of the transactional database that focuses on the products for spreading and the geographical zones proposed in these estimates.

A spreading project developed by a research consultancy contains information about the products to be spread. With a view towards traceability, the products are grouped together in batches (“Batch of product for spreading” class); a batch corresponds to the organic matter from a given producer for a given period. For each batch of products, the different spreadings for which provision has been made are specified (“Spreading” class). For each spreading, the geographical zone on which

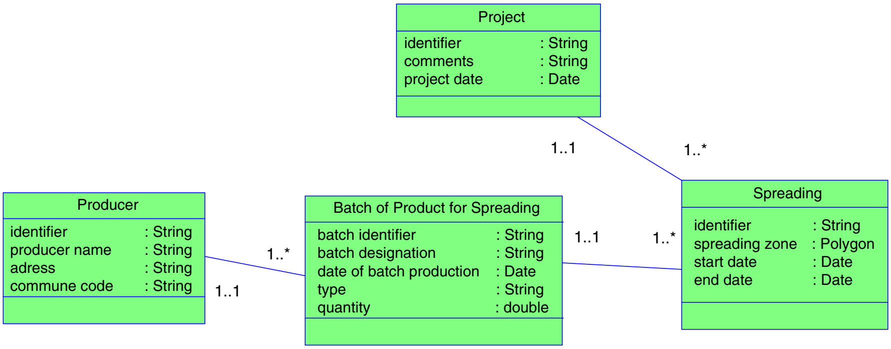

**Fig. 3** Conceptual data model

the processing is to take place is specified (“spreading zone” attribute). A spreading zone is represented by a polygon. The period during which the spreading is to take place is established by the “start date” and “end date” attributes. The organic matter is classified according to types (“type” attribute of the “Batch of product for spreading” class), for example, effluent from livestock or sludge from water treatment plants. Precise information about the producer is also given (“Producer” class). Figure 4 shows examples of batch spreading in different zones. Each zone is stored in the value of the “spreading zone” attribute of an instance of the “Spreading” class. These geographical spreading zones may overlap and even be exactly equal. For example, there is an overlapping area for Zone 2 and Zone 3.

## 3 UML profile

In our method, UML class diagrams describe multi-dimensional models. Our profile provides a conceptual formalism based on UML to build a multi-dimensional model of a data warehouse. The novelty of our approach is that classes can represent both a fact and the level of a dimension (Sect. 3.4). This makes it possible to describe different categories of analysis focusing on the same data.

Here we show the main concepts of our UML profile.

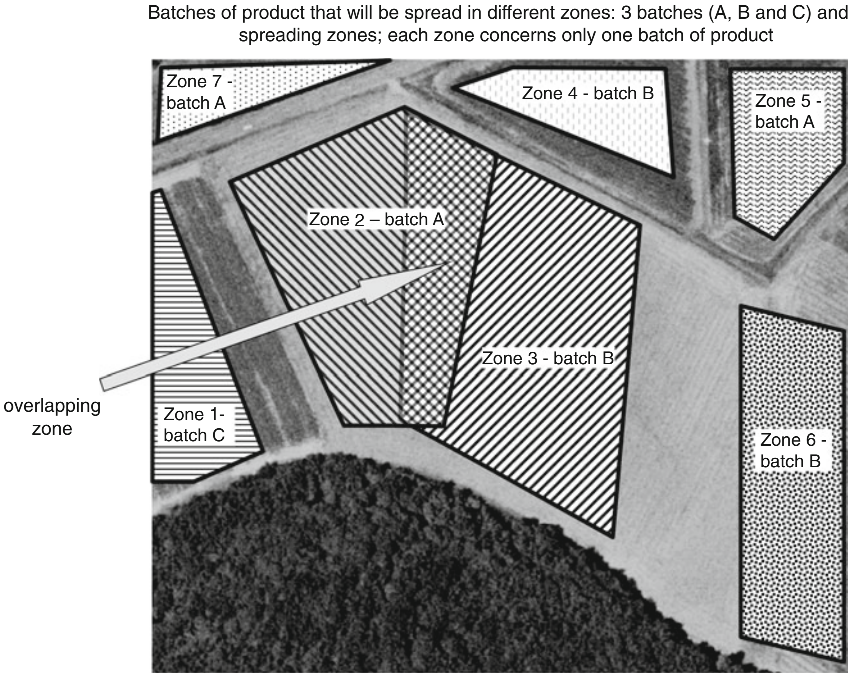
Batches of product that will be spread in different zones: 3 batches (A, B and C) and 7 spreading zones; each zone concerns only one batch of product
Zone 7 - batch A
Zone 4 - batch B
Zone 5 - batch A
Zone 2 – batch A
Zone 3 - batch B
overlapping zone
Zone 1- batch C
Zone 6 - batch B

**Fig. 4** Example of data

## 3.1 Class of facts

A class of facts is a class containing measure attributes on which the analyses focus. An instance of this type of class is called a *fact*. In our UML profile, a class of facts is given on a diagram by the tagged value {fact}. The measure attributes for the analyses are tagged by {m}. {id} shows the attributes that identify the class.

For example, suppose we wish to analyze the surface area of spreadings by calculating the spreading areas by batch of products or by producer. The first stage will be to specify the class of facts in the multi-dimensional model, as shown in Fig. 5. This derives from the corresponding class of the model in Fig. 3.

## 3.2 Class of members

In our multi-dimensional models, a *dimension* is a hierarchy of classes of members; each of these classes describes a level of analysis in the dimension. An instance of this type of class is called a *member*. Classes of members are directly or indirectly linked by associations with at least one class of facts. In our profile, these associations are called aggregating associations. An aggregating association is an oriented link "Source → Destination" that makes it possible to show that an aggregation function (sum, average, spatial union etc.) can be used on separate groups of instances of Source class, so as to give an indicator value for each instance of the Destination class. For example, by using spatial union as an aggregation function, it is possible to determine an overall geographical zone for each project that corresponds to the spreadings for the project (and to calculate its surface area in hectares, for example). In this way, an aggregating association can be established between "Spreading" and "Project". The spatial union is a traditional function used in Geographical Information Systems (for example, see Manolopoulos et al. 2004). It merges several geometries into one; the resulting geometry is a single object that may or may not consist of several disconnected shapes.

We will now specify how to produce a multi-dimensional model using the model of a transactional database, and explain the semantics of multi-dimensional models.

### *3.2.1 Changing from a transactional database to a multi-dimensional model*

Using the model in Fig. 3, it is possible to show the dimensions of possible analyses relatively easily. Continuing on from the previous example, we want to calculate the surface area of spreadings according to different dimensions. The possible classes of members can be fairly naturally obtained using the existing one-to-many

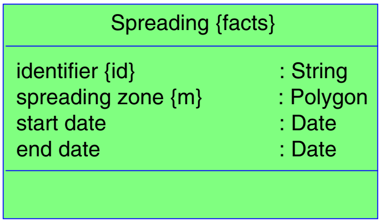

**Fig. 5** "Spreading" class of facts

associations of the model presented in Fig. 3 (Tsois et al. 2001). Indeed, a dimension can correspond to a hierarchy connected to the class of facts called "Spreading" and made up of classes linked to each other by one-to-many associations. In a hierarchy, each class is a possible analysis level; it can therefore be a class of members in the multi-dimensional model. Figure 6 shows all the classes of members; the one-to-many associations in Fig. 3 have been replaced by aggregating associations noted by "←" in the diagram of Fig. 6.

### *3.2.2 Semantics of a multi-dimensional model*

The different possible dimensions correspond to hierarchies of classes that "start" from the "Spreading" class ("spreading zone" is the measure attribute). So, the model shows that a spreading zone can be analyzed according to two dimensions: a "Project" dimension and a "Batch of Product for Spreading → Producer" dimension. We can do analyses by combining the levels of different dimensions. For example, take spatial union as the aggregation function to apply to the measure attribute of the class of facts. In this case, the model shows that it will be possible to specify the geographical spreading zones:

- by project,
- by producer,
- by batch of spreading products,
- by project and producer,
- by project and batch of spreading products.

We can then calculate the surface area (in hectares, km$^2$, etc.) of the geographical zones determined by this procedure. For this calculation, we can use a specialized function. According to requirements, the classes and attributes of Fig. 3 that cannot be used for analysis may not be shown in the multi-dimensional model.

For the analysis dimensions specified here, Fig. 7 gives unions of zones resulting from calculations obtained for the data of Fig. 4. In this example, we assume that:

- zones 1, 2 and 3 come from a PrjI project,
- zones 4, 5, 6 and 7 come from a PrjJ project,

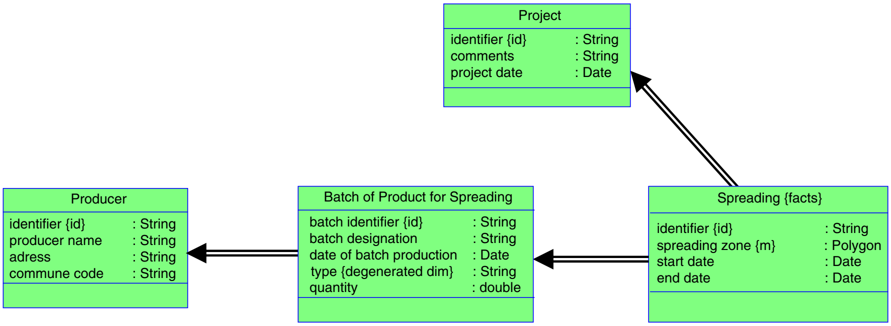

**Fig. 6** Multi-dimensional conceptual model

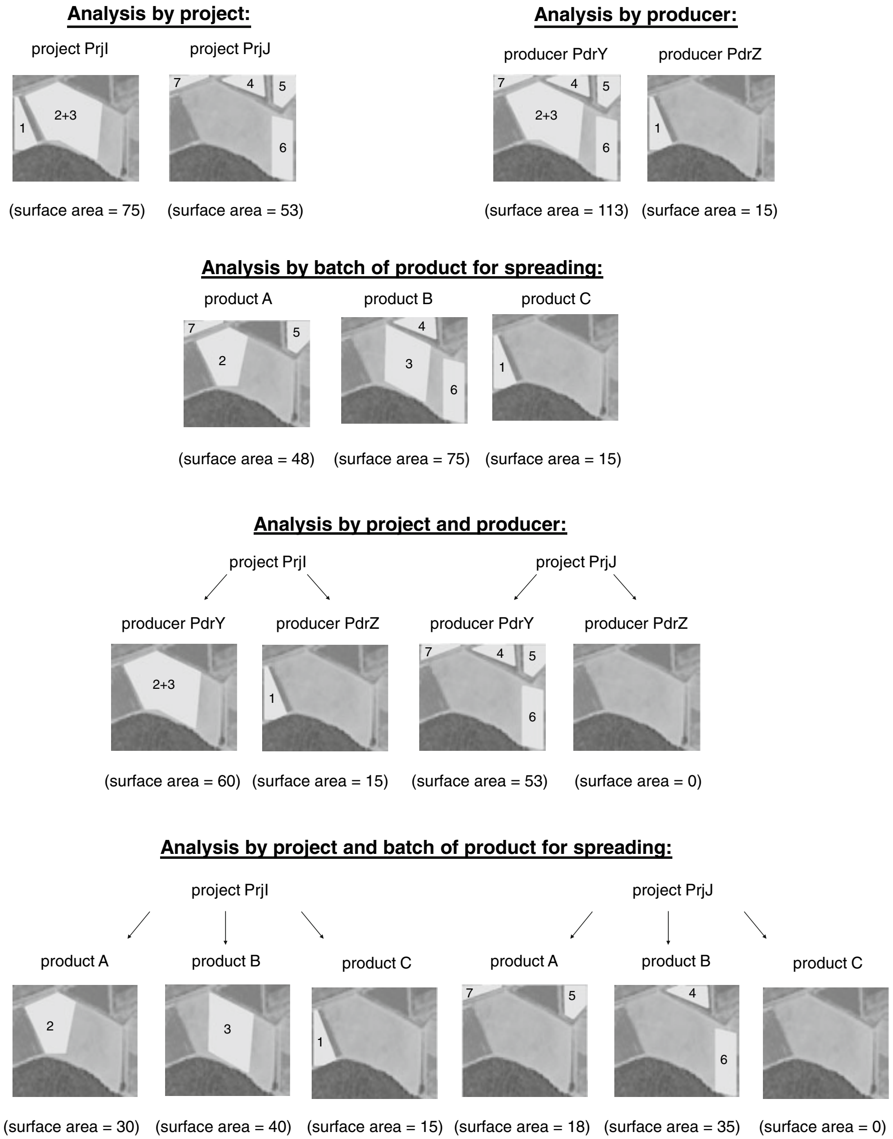

**Fig. 7** Analysis of spreading zones: results taking into consideration the analysis dimensions of Fig. 6

- batches A and B of products come from the same PrdY producer,
- batch C of products comes from the PrdZ producer.

Table 1 shows the surface areas of each zone.

For the analysis by project, the geographical zones are grouped by projects (PrjI and PrjJ). For the analysis by project and producer, a union of zones is calculated for each couple {project, producer}, etc. We can see from the results that the sum of surface areas obtained by analysis is either 128 or 138. For example, the sum of

**Table 1** Surface areas of each zone

| Zone | Surface area | Producer | Project | Batch |
|---|---|---|---|---|
| 1 | 15 | PrdZ | PrjI | C |
| 2 | 30 | PrdY | PrjI | A |
| 3 | 40 | PrdY | PrjI | B |
| 4 | 15 | PrdY | PrjJ | B |
| 5 | 10 | PrdY | PrjJ | A |
| 6 | 20 | PrdY | PrjJ | B |
| 7 | 8 | PrdY | PrjJ | A |

surface areas is 138 for the analysis by batch of product (48 + 75 + 15) and 128 for the analysis by project (75 + 53). This difference is because geographical zones 2 and 3 overlap. The surface area of the spatial union of these two zones is less than the sum of their surface area. In our example, we consider that the surface area of the union of the zones 2 and 3 is 60.

Moreover, all the attributes in the classes of the model can be used to produce selections limiting the data for analysis. For example, it is possible to use the "start date" and "end date" attributes so as to only take the geographical zones of a certain period into consideration in the calculations.

Note that in our profile, a class of members can be indicated in the diagram by a tagged value {members}. However, this indication remains optional in the diagrams. A class in the hierarchy of a dimension is always a class of members, even if it does not have the tagged value {members} in the diagram.

### 3.3 Degenerated dimension

Certain attributes may in fact correspond to levels of a dimension. We call these attributes "degenerated dimensions" (Malinowski and Zimanyi 2008). An example is the "type" attribute in the "Batch of product for spreading" class. The type of product may correspond to a possible level of analysis. In our profile, these attributes may be tagged {degenerated dim} (as in Fig. 6) or may appear in the form of a class of members. Figure 8 shows this latter mode of representation. We can therefore see that it is now possible to produce analyses taking into account product categories: for example, calculating geographical spreading zones by project, producer and category of product. Other attributes may also be used as a degenerated dimension. This is the case, for example, for the "commune code" attribute of the "Producer" class.

### 3.4 Unified representation of facts and members: a method for improving the flexibility of models

Using a model of a transactional database, it is possible to produce different analyses focusing on different dimensions. As we shall demonstrate here, a class of members of an analysis may very well be a class of facts for another analysis. Using

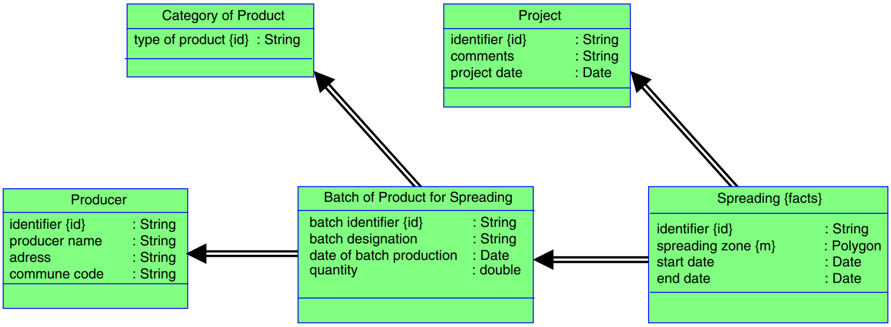

**Fig. 8** Canceling the "type" attribute from the "Batch of product for spreading" class and creating a new class of members called "Category of product"

our approach, it is possible to specify a model showing several categories of analysis at the same time.

Let us suppose that we want to analyze the quantities of batches. In this case, "Batch of product for spreading" is then a class of facts. The usable dimensions for this analysis are the ones that "start" from this class (see Fig. 9). We can see here that it will then be possible to analyze the spreading products (for example, calculate the aggregate quantities of products) by category of product or producers. "Batch of product for spreading" is also a class of members as it still corresponds to a level of analysis of "Spreading".

In our approach, the facts and the members are represented in the same way (by a class). So, within a single model, a class of facts may also be a class of members. This allows great flexibility, as a model can describe several categories of analyses, each focusing on different measure attributes, while sharing common dimensions.

So, we have proposed an easy-to-use profile that makes it possible to describe the main concepts used in the multi-dimensional models (such as facts, members, and measure). Figure 10 shows the part of the meta-model of the UML that our profile has extended. The classes belonging to our profile are labeled "stereotype". The other classes make up part of the meta-model of the UML. In the diagram, we use the same arrangement of classes as the one used to describe the profile shown by

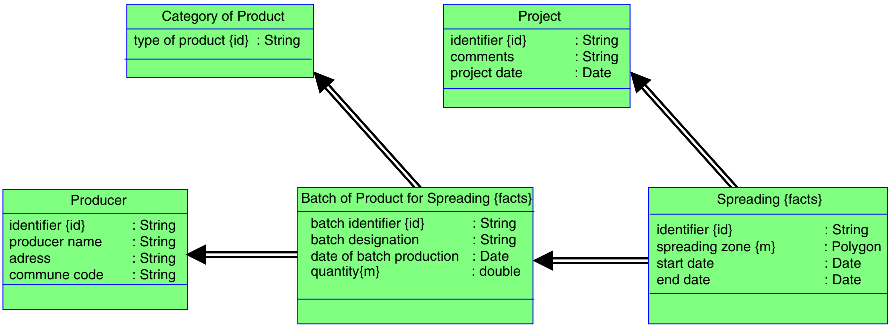

**Fig. 9** A new multi-dimensional model for analyses by quantities or geographical zones

Mazon and Trujillo (2008). In Fig. 10, the aggregating associations specialize traditional associations. Classes of facts and classes of members inherit all the features from a traditional class. Degenerated dimensions, measurements (measures) and id are special attributes.

The profile that we propose is still compatible with the profiles and Entity-Relation extensions proposed in the field (Lujan-Mora et al. 2006; Malinowski and Zimanyi 2006, 2008; Prat et al. 2006; Mazon and Trujillo 2008); our proposal could therefore easily be supplemented by additional concepts introduced by these profiles. The novelty of approach stems from the possibility of considering several categories of analyses on a single multi-dimensional model (by allowing a class of facts to be a class of members).

## 4 Integrating several databases

### 4.1 Data integration

In principle, a data warehouse integrates several different databases or data sources for analysis. Here, we shall show how to view several data sources in a single multi-dimensional model.

Let us look at data on the administrative division of France's territory, shown in Fig. 11. France's territory is made up of communes (urban or rural towns) that are grouped together in departments (or counties), which themselves are grouped together into regions. There are 26 regions, 100 departments and more than 36,000 communes.

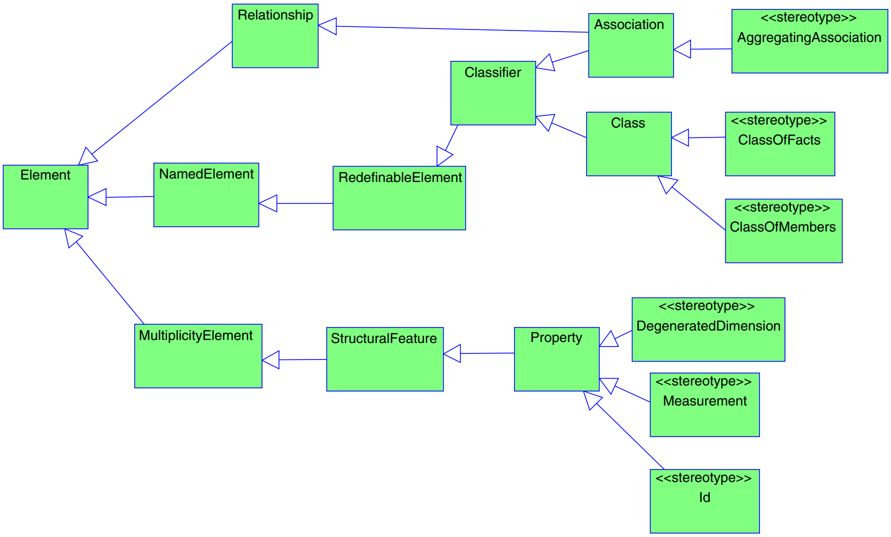
Relationship
Association
<<stereotype>> AggregatingAssociation
Classifier
Class
<<stereotype>> ClassOfFacts
Element
NamedElement
RedefinableElement
<<stereotype>> ClassOfMembers
<<stereotype>> DegeneratedDimension
MultiplicityElement
StructuralFeature
Property
<<stereotype>> Measurement
<<stereotype>> Id

**Fig. 10** UML profile for multi-dimensional models

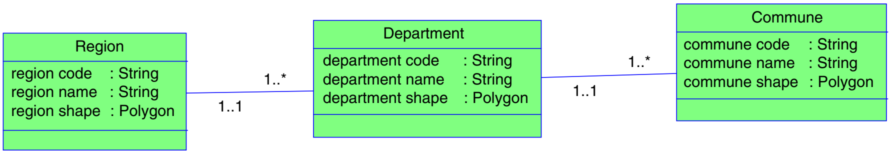

**Fig. 11** Data model for administrative divisions of France's territory

Within a data warehouse, it is possible to integrate this information with information from the spreadings database. This will make it possible to produce certain spatialized analyses. Figure 12 shows the new multi-dimensional model. The one-to-many associations of the model presented in Fig. 11 give new aggregating associations in the multi-dimensional model. A spreading zone may be associated with several communes, and a commune may contain several spreading zones. This is shown in Fig. 13.a. So, for example, to calculate the surface areas of spreading zones by department and region, the spreading zones by commune must first be specified (see Fig. 13b). This requires specific spatial processing when integrating the data source of Fig. 3 with that of Fig. 11; the results of these calculations are stored in the data warehouse using the "zone by commune" attribute of the new class of facts called "Spreading by commune".

It is no longer necessary for "Spreading" to be a class of facts. This becomes a class of members in the dimensions: "Spreading → Batch of Product for Spreading → Producer", "Spreading → Batch of Product for Spreading → Category of Product" and "Spreading → Project".

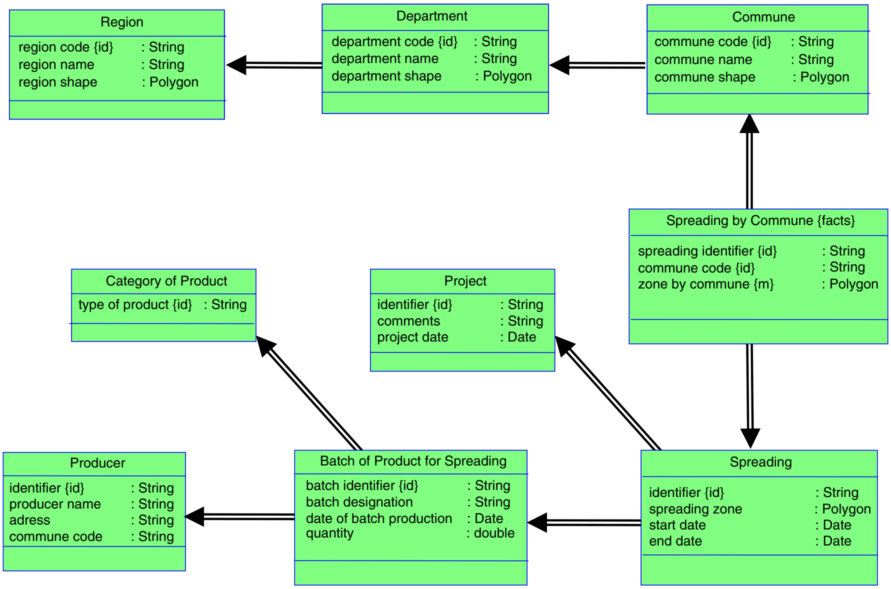

**Fig. 12** Integrating a spatial dimension

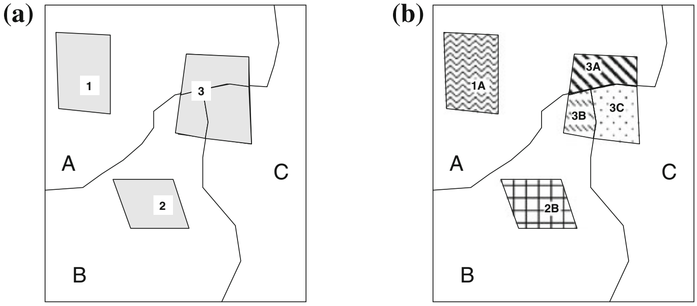

**Fig. 13** Calculation of spreading zones by commune. **a** Three spreading zones (1, 2, 3) and the border between three communes (*A*, *B*, *C*). Zone 3 overlaps the three communes, **b** Calculation of the zones by commune—now there are five zones (1A, 2B, 3A, 3B, 3C). Each zone concerns only one commune

Let us continue to consider spatial union as a function of aggregation. The multi-dimensional model of Fig. 12 shows that it will then be possible to determine the spreading zones (and their surface area):

- by project and commune,
- by project and department,
- by project, category of product, producer and commune, etc.

Table 2 shows these new spatialized analyses. With this example, it is clear that adding classes of members to the model multiplies the number of possible analyses.

## 4.2 Integrity of data

Certain integrity rules can be checked so as to make sure that no errors have occurred during data integration. For example, during construction, it should not be possible for spreading zones by commune to go over the borders of their commune. Zone 3A of Fig. 13b do not go over the borders of commune A; if this were not the case, an error in the zone integration or calculation process would be suspected.

Modeling and checking integrity constraints are important because they partially affect the quality of data. Poor quality can make analysis results unreliable and thus can also lead to poor decisions. The larger the number of sources integrated, the more elusive quality becomes. Some example causes of poor quality are:

- merging sources corresponding to different periods,
- poor quality of a data source (which will be reflected in the data warehouse),
- errors (bugs) in the data integration algorithms,
- differing data precision levels in different sources.

Currently, no proposals have been made for specifying the integrity constraints in the data warehouses with a formal language. In multi-dimensional models, we argue for modeling these constraints in OCL, the UML language of constraints (OMG 2005; Schmid et al. 2002). This makes it possible both to document the models unambiguously in this standard recognized language and to automatically generate

**Table 2** Different spatialized analyses issued from the model of Fig. 12

| Number of criteria | Analysis |
|---|---|
| 1 | By (commune); by (department); by (region) |
| 2 | By (project and commune); by (project and department); by (project and region) |
| | By (batch of spreading product and commune) |
| | By (batch of spreading product and department) |
| | By (batch of spreading product and region); by (producer and commune) |
| | By (producer and department); by (producer and region) |
| | By (category of product and commune); by (category of product and department) |
| | By (category of product and region) |
| 3 | By (category of product, producer and commune) |
| | By (category of product, producer and department) |
| | By (category of product, producer and region) |
| | By (project, batch of spreading product and commune) |
| | By (project, batch of spreading product and department) |
| | By (project, batch of spreading product and region) |
| | By (project, producer and commune); by (project, producer and department) |
| | By (project, producer and region); by (project, category of product and commune) |
| | By (project, category of product and department) |
| | By (project, category of product and region) |
| 4 | By (project, category of product, producer and commune) |
| | By (project, category of product, producer and department) |
| | By (project, category of product, producer and region) |

the computing code that will check the data. There are many techniques and tools for generating data verification mechanisms using constraints expressed in OCL (Demuth and Hußmann 1999; Demuth et al. 2001, 2004; Demuth 2005; Pinet et al. 2005, 2007, 2009; Klasse Objecten 2008).

Using UML class diagrams to represent data warehouses lets us use OCL directly. The following example shows how OCL integrity rules can be specified in multi-dimensional models. This OCL constraint uses a variant of spatial operations introduced in (Duboisset et al. 2005; Pinet et al. 2007, 2009). It shows that the shape of the spreading zone by commune (zone_by_commune) must either equal, be within or be covered by the shape of the commune (commune_shape). Egenhofer and Franzosa (1991); Egenhofer and Herring (1992) give precise definitions of these topological relationships.

```
context Spreading_by_Commune inv:
(self.zone_by_commune).{equal|inside|coveredBy}
(self.Commune.commune_shape)
```

Let us now assume that we wish to produce a data warehouse devoted to analyzing spreadings for a given year. The data that will be entered into the data warehouse must therefore only concern the relevant year. Therefore, only certain instances from the database in Fig. 3 will be integrated within the data warehouse, these instances being the ones that correspond to the analysis period. The following constraint will make it possible to verify whether the spreading zone actually concerns the period being examined (in the example given here, this is 2008); the validity start date must be 2008/1/1 or after, and the validity end date must be 2008/12/31 or before.

```
context Spreading inv:
self.start_date > = 20080101 and
self.end_date <= 20081231
```

Let us now look at a new data source showing the list of farms. We can integrate this new source within the data warehouse in Fig. 12. If all farm producers are farmers during the entire period under consideration, we can introduce two new classes of members: “Farmer” and “Other Types of Producers”. Figure 14 shows this solution; the figure only shows classes that have been modified from Fig. 12. A batch of product is therefore associated either with an instance of “Farmer” or with an instance of “Other Types of Producers”. The union of instances of these two classes corresponds to all instances of the “Producer” class. The multi-dimensional model therefore indicates that it is possible to produce analyses by farmer, by other types of producers or even by any type of producers. Constraints can be established in this model. For example, a batch of product comes from either a farmer or another type of producer. This constraint can be specified visually directly in the diagram (see Fig. 14) or using the following OCL expression:

```
context Batch_of_Product_for_Spreading inv:
self.Farmer- > notEmpty() xor
self.Other_Types_of_Producers- >notEmpty()
```

The latter constraint is important, as it implies that it is unnecessary to combine Farm and Other types of producers in the same analysis at the same time. For example, the quantities of product batches calculated by farmer AND by other types of producer will always be zero.

Other constraints can be expressed. For example, the group of instances of “Producer” class is the union of instances of “Farmer” and “Other Types of Producers” classes:

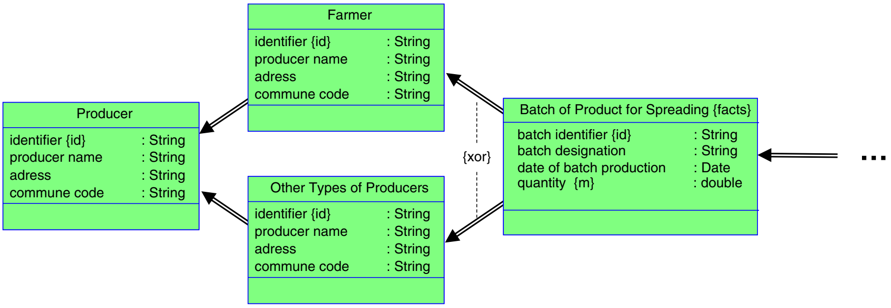

**Fig. 14** Two new classes of members

```
context Producer inv:
Producer.allInstances() =
Farmer.allInstances()- > union
(Other_Types_of_Producers.allInstances())
```

The product batches of type “effluent from farm livestock” must come from a farm:

```
context Batch_of_Product_for_Spreading inv:
if self.Category_of_Product.type_of_product = ‘effluent from
farm livestock’
then self.Farmer.notEmpty()
```

# 5 Using the data

## 5.1 Implementing the multi-dimensional model

Users can examine warehoused data interactively using OLAP (On-Line Analytical Process) tools (Malinowski and Zimanyi 2008). Among other things, these tools allow you to view the results of different analyses by selecting the levels of the dimensions required or by limiting the members involved in an analysis. The main results are sometimes calculated once and once only; they are stored “permanently” directly in the data warehouse, enabling very fast access when examining data.

The main types of OLAP tools are:

- relational OLAPs (ROLAP); a ROLAP tool accesses data from the data warehouse that are stored in a relational database. Using this approach, the multi-dimensional model needs to be converted to a relational model. The two types of relational implementation that are most used are the star schema and the snowflake schema.
- multi-dimensional OLAPs; a MOLAP tool accesses data directly stored in the form of cubes with *n*-dimensions that combine the different levels of analysis. Figure 15 shows an example of a cube corresponding to analysis by product category, producer and commune (see model in Fig. 12). Each cell therefore corresponds to the spreading zone calculated for a product category, a producer and a commune. Certain cells may be empty. {commune 1,…, communes *u*} are members of the “Commune” level. {producer 1,…, producers *v*} are members of the “Producer” level. The members of the “Category of Product” level are {effluent from farm livestock, effluent from food industry,…, sludge from water treatment plant}.
- hybrid OLAPs (HOLAP); a HOLAP tool accesses data stored in relational databases and in cubes with *n*-dimensions.

These enable users to run specific queries; for example, if we take the data in Fig. 15:

- select spreading zones from all the communes of a given department,
- find for each product category all communes with a spreading area higher than a given value,

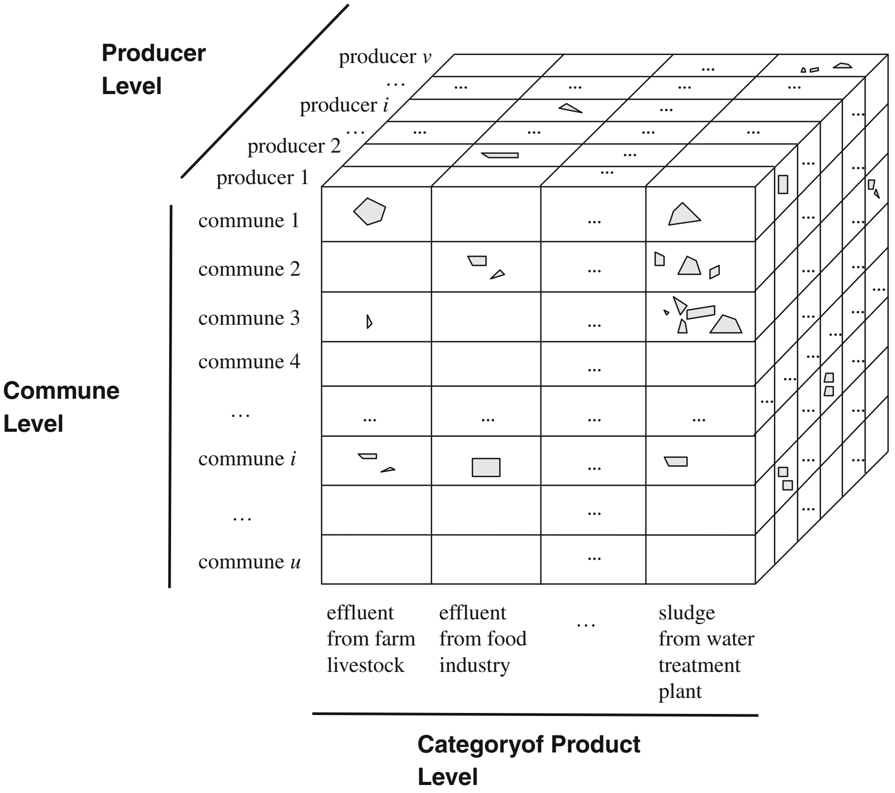

**Fig. 15** Cubes corresponding to calculating spreading zones by category of product, producer and commune

- select producers supplying over 80% of the total spreading area of at least one commune.

The results of these queries can be new cubes.

Currently, some spatial extensions have been proposed for OLAPs so as to view the geographical data of data warehouses on maps (Bimonte et al. 2006; McHugh et al. 2009). These tools are of significant interest in fields such as agriculture and the environment.

## 5.2 Data-mining and optimization

OLAP tools can help people make decisions. Decision-makers can examine the different analyses supplied by the data warehouse to get a complete picture of the situation. Within the environmental and agricultural fields, integrating sources of different data should help with decision-making and optimization. The spatial dimension continues to be vital for many decisions: for example, finding a place for installing infrastructures, choosing the best positions for cultivation, or optimizing the organization of farming practices by time and space. Optimization algorithms can use data from cubes. For example, using data from the data warehouse presented in this article, it is possible to look for the best positions for a new infrastructure by trying to avoid, as much as possible, pollution associated with spreading some materials. In addition, this decision may be made by integrating other constraints (such as surroundings or pollution). This will mean that other information will need to be integrated within the data warehouse. Similarly, using data from certain cubes, it will be possible to optimize farming practices by determining precisely how spreading operations are organized over time. This optimization will mean that such issues as products, producers, geographical zones to be treated, and product categories will need to be taken into account.

Data-mining algorithms can be used in data warehouses, still with a view toward improving data analysis. By combining data from different sources, it is possible to show the correlations between objects dealt with, for example, to show the links between a type of industrial installation and disease. The results of these algorithms can then be stored and shown to decision-makers with the rest of the information from the data warehouse. Berson and Smith (1997), among others, have summarized the application of data-mining to data warehouses.

## 6 Conclusion and prospects

While use of data warehouses is becoming widespread in certain spheres of activity such as finance and mass marketing, its use in environmental and agricultural fields is still marginal. This technology may, however, be of great service in these sectors, whether in decision-making or in optimization of agricultural and environmental systems. This paper first provides an example of a data warehouse application for a real agricultural scenario; we explain iterative integration of different information sources. It then addresses modeling research problems further upstream.

We have introduced a UML profile that makes it possible to specify multi-dimensional models of data warehouses. Compared with the formalisms currently available, the added value of using the proposed profile consists of: (a) the ability to indicate several categories of analyses in a single model (see Sect. 3.4), (b) the use of OCL to model integrity constraints in data warehouses (Sect. 4.2).

Describing several categories of analyses in a single diagram is vital when a single multi-dimensional structure needs to be reused for different analysis objectives. The additional variations of existing UML profiles for data warehouses can easily be integrated into our proposal. The profile presented here is thus still compatible with existing profiles. We illustrated our profile with the example of an environmental data warehouse. Nevertheless, our proposal is generic: the profile can be used in any other context of conception of data warehouses.

Using OCL to model integrity constraints is an original approach within the context of data warehouses. Problems with establishing data quality checking procedures are practically un-addressed in the data warehouse literature.

Our aim is to continue the processes described here and implement them in order to propose software tools for automatically generating data storage structures using UML models and automatically producing mechanisms for checking OCL integrity constraints in data warehouses. The generation of data storage structures can be done by adapting existing Case Tools (based on UML). In the environmental field, it will be vital to pay more notice to modeling and storing geographical data. To this end, the proposals of Malinowski and Zimanyi about multi-dimensional modeling (2008) may be considered, together and more generally with UML extensions for Geographical Information Systems (Brodeur et al. 2000; Friis-Christensen et al. 2001; Miralles and Libourel 2007; Pinet et al. 2005). The automatic production of mechanisms for checking OCL integrity constraints in data warehouses will make it possible to compensate for the current lack of concrete ways to check the quality of integrated data.

As well as those presented in this section, the prospects for research in the field of UML modeling of data warehouses are still numerous and varied; see Rizzi et al. for other examples of prospects (2006). There are still a great many obstacles, but we think that UML will be of great service in the implementation of data warehouses, including the integration (Trujillo and Luján-Mora 2003) and data-mining phases (Zubcoff and Trujillo 2007).

## References

Berson A, Smith S (1997) Data warehousing, data mining, and OLAP (Data Warehousing/Data Management), Computing Mcgraw-Hill (640 p)

Bimonte S, Tchounikine A, Miquel M (2006) GeoCube, a multidimensional model and navigation operators handling complex measures: application in spatial OLAP. Lect Notes Comput Sci 4243:100–109

Brodeur J, Bedard Y, Proulx MJ (2000) Modelling geospatial application databases using UML-based repositories aligned with international standards in geomatics. Proceedings of the international ACM symposium on advances in geographic information systems, USA, pp 39–46

Calì A, Lembo D, Lenzerini M, Rosati R (2003) Source integration for data warehousing. Multidimensional databases, pp 361–392

Cemagref (2008) <http://www.cemagref.fr/english>

Demuth B (2005) The dresden OCL toolkit and the business rules approach. European business rules conference (EBRC 2005), Amsterdam

Demuth B, Hußmann H (1999) Using UML/OCL constraints for relational database design. Lect Notes Comput Sci 1723:598–613

Demuth B, Hußmann H, Loecher S (2001) OCL as a specification language for business rules in database applications. Lect Notes Comput Sci 2155:104–117

Demuth B, Loecher S, Zschaler S (2004) Structure of the dresden OCL toolkit. In: 2nd Interna-tional Fujaba days “MDA with UML and rule-based object manipulation”. Darmstadt, Germany, September 15–17

Duboisset M, Pinet F, Kang MA, Schneider M (2005) Precise modelling and verification of topological integrity constraints in spatial databases: from an expressive power study to code generation principles. Lect Notes Comput Sci 3716:465–482

Egenhofer M, Franzosa R (1991) Point-set topological spatial relations. Int J Geogr Inf Syst 5(2):161–174

Egenhofer M, Herring J (1992) Categorizing binary topological relationships between regions, lines, and points in geographic databases. Technical report. Department of surveying engineering. University of Maine, Orono, 28 p

Friis-Christensen A, Tryfona N, Jensen C (2001) Requirements and research issues in geographic data modelling. In: Proceedings of the international ACM symposium on advances in geographic information systems, USA, pp 2–8

Klasse Objecten (2008) OCL tools web site http://www.klasse.nl/ocl

Lujan-Mora S, Trujillo J, Song IY (2006) A UML profile for multidimensional modelling in data warehouses. Data Knowl Eng 59(3):725–769

Malinowski E, Zimanyi E (2006) Hierarchies in a multidimensional model: from conceptual modelling to logical representation. Data Knowl Eng 59(2):348–377

Malinowski E, Zimanyi E (2008) Advanced data warehouse design: From conventional to spatial and temporal applications, Springer (435 p)

Manolopoulos Y, Papadopoulos A, Vassilakopoulos M (2004) Spatial databases: technologies, techniques and trends, IGI Global (340 p)

Mazon JN, Trujillo J (2008) An MDA approach for the development of data warehouses. Deci Support Syst 45:41–55

McHugh R, Roche S, Bedard Y (2009) Towards a SOLAP-based public participation GIS. J Environ Manage 90(6):2041–2054

Miralles A, Libourel T (2007) Spatial database modelling with enriched model driven architecture. Encyclopedia of geographical information sciences, Springer (9 p)

Muzy A, Innocenti E, Aïello A, Santucci JF, Santoni PA, Hill D (2005) Modelling and simulation of ecological propagation processes: application to fire spread. Environ Modell Softw 20(7):827–842

Nilakanta S, Scheibe K, Rai A (2008) Dimensional issues in agricultural data warehouse designs. Comput Electron Agric 60(2):263–278

OMG (2005) OMG: OCL 2.0 specification (185 p)

Papajorgji P (2007) State of the art in modeling software for agricultural systems. Encyclopedia of optimization, second edition, Springer

Papajorgji P, Pardalos P (2006) Software engineering techniques applied to agricultural systems: an object-oriented and UML approach. Springer (247 p)

Papajorgji P, Shatar P (2004) Using the unified modelling language to develop soil water-balance and irrigation-scheduling models. Environ Modell Softw 19(5):451–459

Papajorgji P, Beck H, Braga J (2004) An architecture for developing service-oriented and component-based environmental models. Ecol Modell 179(1):61–76

Pinet F, Kang MA, Vigier F (2005) Spatial constraint modelling with a GIS extension of UML and OCL: application to agricultural information systems. Lect Notes Comput Sci 3511:160–175

Pinet F, Duboisset M, Soulignac V (2007) Using UML and OCL to maintain the consistency of spatial data in environmental information systems. Environ Modell Softw 22(8):1217–1220

Pinet F, Duboisset M, Demuth B, Schneider M, Soulignac V, Barnabé F (2009) Constraints modelling in agricultural databases. Chapter In: Advances in modeling agricultural systems

Prat N, Akoka J, Comyn-Wattiau I (2006) A UML-based data warehouse design method, Decision support systems 42(3), pp 1449–1473

Rizzi S, Abello A, Lechtenborger J, Trujillo J (2006) Research in data warehouse modelling and design: dead or alive? Proceedings of the 9th ACM international workshop on data warehousing and OLAP, pp 3–10

RN DEAS (2008) Research network "Design of environmental & agricultural systems", <http://deas.research.free.fr>

Schmid B, Warmer J, Clark T (2002) Object modelling with the OCL: the rationale behind the object constraint language, springer, (281 p)

Schneider M (2008) A general model for the design of data warehouses. Int J Prod Econ 112(1):309–325

Schulze C, Spilke J, Lehner W (2007) Data modelling for precision dairy farming within the competitive field of operational and analytical tasks. Comput Electron Agric 59(1):39–55

Soulignac V, Gibold F, Pinet F, Vigier F (2005) Spreading matter management in France within sigemo. In: Proceedings of the 5th European conference for information technologies in agriculture (EFITA 2005), Vila Real, Portugal, July 25–28 (5 p)

Trujillo J, Luján-Mora S (2003) A UML based approach for modelling ETL processes in data warehouses. Lect Notes Comput Sci 2813:307–320

Trujillo J, Palomar M, Gomez J, Song IY (2001) Designing data warehouses with OO conceptual models. IEEE Comput 34(12):66–75

Tsois A, Karayannidis N, Sellis T (2001) MAC: conceptual data modelling for OLAP. In: Proceedings of the international workshop on design and management of data warehouses, Interlaken, Switzerland

Zubcoff JJ, Trujillo J (2007) A UML 2.0 profile to design association rule mining models in the multidimensional conceptual modelling of data warehouses. Data Knowl Eng 63(1):44–62