

This article was downloaded by: [Kamal BOULIL]

On: 12 November 2014, At: 07:49

Publisher: Taylor & Francis

Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



**Journal
of Decision
Systems**

Volume 23 Number 4 November 2014

Editorial Board
Managing Editor
Production Editor

Journal of Decision Systems

Publication details, including instructions for authors and subscription information:

<http://www.tandfonline.com/loi/tjds20>

Spatial OLAP integrity constraints: From UML-based specification to automatic implementation: Application to energetic data in agriculture

Kamal Boulil^a, Sandro Bimonte^a & Francois Pinet^a

^a Irstea, UR TSCF, 24 avenue des Landais, 63172 Aubière, France

Published online: 13 Oct 2014.



To cite this article: Kamal Boulil, Sandro Bimonte & Francois Pinet (2014) Spatial OLAP integrity constraints: From UML-based specification to automatic implementation: Application to energetic data in agriculture, Journal of Decision Systems, 23:4, 460-480, DOI: [10.1080/12460125.2014.934120](https://doi.org/10.1080/12460125.2014.934120)

To link to this article: <http://dx.doi.org/10.1080/12460125.2014.934120>

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the "Content") contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms & Conditions of access and use can be found at <http://www.tandfonline.com/page/terms-and-conditions>

Spatial OLAP integrity constraints: From UML-based specification to automatic implementation: Application to energetic data in agriculture

Kamal Boulil*, Sandro Bimonte and Francois Pinet

Irstea, UR TSCF, 24 avenue des Landais, 63172 Aubière, France

(Received 30 September 2012; accepted 12 April 2014)

Spatial OnLine Analytical Processing systems (SOLAP) are Business Intelligence technologies allowing efficient and interactive analysis of large spatial data cubes. In this type of systems, the correctness of analysis depends on the warehoused data quality, how aggregations are performed and how data cubes are explored. In this paper, we study quality control techniques (based on integrity constraints) related to exploration of spatial data cubes. We extend our Unified Modeling Language (UML) framework previously proposed with a UML profile allowing the conceptual design of several classes of exploration integrity constraints. We also propose a tool for their automatic implementation. We validate our proposal in the context of a real case study concerning the analysis of energetic farm indicators.

Keywords: spatial data warehouse; spatial OLAP; quality control; UML; OCL; integrity constraints

1. Introduction

Currently, the number of available geo-referenced datasets is increasing due to new spatial data acquisition technologies (e.g. sensor networks, remote sensing techniques, etc.). Spatial Database Management Systems (SDBMS), such as PostGIS, Oracle, etc., provide a native support for storage and querying of spatial data. On the other hand, complex spatial analysis and visualisation methods are provided by means of Geographic Information Systems (GIS), such as ArcGIS, QGIS, etc. In this context, to exploit the advanced analysis capabilities of this geographic data, Business Intelligence (BI) technologies such as spatial data mining, Spatial OnLine Analytical Processing systems (SOLAP) and Spatial Data Warehouses (SDW) have been proposed. In particular, SDW and SOLAP provide online and spatio-multidimensional analyses of geo-referenced information with consideration of spatial components (e.g. the geometry) (Bédard, Rivest, & Proulx, 2006). SDW has been defined as ‘A collection of subject-oriented, integrated, nonvolatile and time-variant spatial and non spatial data to support the decision-making process’ (Stefanovic, Han, & Koperski, 2000). Warehoused data are modelled according to the spatio-multidimensional model, which defines the concepts of spatial dimension (geo-referenced analysis axes), and spatial measures (geo-referenced measures). Spatial measures are aggregated along dimension hierarchies using specific aggregate functions (Malinowski & Zimányi, 2008), and warehoused

*Corresponding author. Email: kamal.boulil@gmail.com

spatial data are analysed by means of SOLAP tools. SOLAP is an approach to answering multi-dimensional analytical queries (once data is stored in SDW). SOLAP has been defined by Yvan Bédard as ‘Visual platforms built especially to support rapid and easy spatiotemporal analysis and exploration of data following a multidimensional approach comprised of aggregation levels available in cartographic displays as well as in tabular and diagram displays’ (Bédard et al., 2006). The SOLAP systems integrate advanced OLAP and GIS functionalities to provide interactive cartographic representations of OLAP query results and use spatial operators for multidimensional query calculations (Bimonte & Pinet, 2010).

Because SDW is able to collect data issued from several heterogeneous data sources, several problems related to quality have impacts on the SOLAP analysis (Boulil, Bimonte, Mahboubi, & Pinet, 2010) and the success of a BI project. The Data Warehousing Institute (TDWI) estimated that up to 611 billion US\$ has been lost because of serious quality problems in BI projects for the US marketplace (Eckerson, 2000). Thus, several works have addressed the data quality in Data Warehouses. In particular, certain approaches have been proposed to ‘repair’ data by means of statistical techniques and data mining techniques, among others (Ribeiro, Goldschmidt, Cláudia, & Cavalcanti, 2011). Other works propose the definition of Integrity Constraints (ICs) to verify the consistency of warehoused data (Salehi, 2009). ICs, rules that express conditions that a data set should satisfy, have been recognised as an effective method to express quality rules for Spatial Databases and SDW (Boulil et al., 2010). Moreover, the quality of spatio-multidimensional analyses also depends on the correct aggregation of measures with respect to summarisability conditions (or aggregation constraints), which check for example that the measures and the aggregate function types are compatible (e.g. the sum of the unit prices does not make sense) (Lenz & Shoshani, 1997). In this paper, we propose techniques for the definition and the control of ICs in SDW that help SDW designers to maintain the desired level of quality. As shown in this paper, the quality control cannot be limited to the two previously described controls (on data and aggregations). The topic also requires additional controls when exploring (aggregated) data (Figure 1) to avoid misinterpretation of meaningless SOLAP query results (Levesque, Bédard, Gervais, & Devillers, 2007).

Thus, in Boulil, Bimonte, and Pinet (2012a, 2012b) we extend the definition of ICs in SOLAP systems and use ICs to perform three control functions:

- (1) Data control ensures that warehoused spatial data are valid and consistent, e.g. the geometries of counties must be topologically included in the geometries of their states;

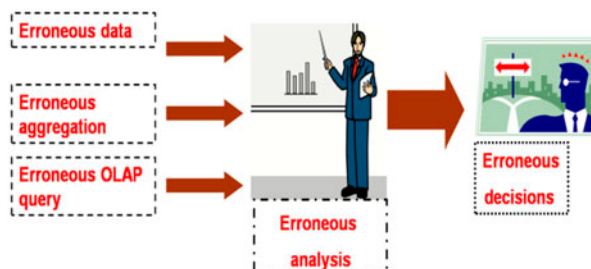


Figure 1. Quality in the SOLAP decision-making process.

Note: OLAP: OnLine Analytical Processing; SOLAP: Spatial OnLine Analytical Processing.

- (2) Aggregation control ensures that aggregations of measures are correct and meaningful, e.g. the sum of temperature values does not make sense (Pedersen, Jensen, & Dyreson, 2001);
- (3) SOLAP exploration control verifies (among other things) the consistency of user queries to avoid problems of misinterpretations of their results, e.g. sales in all countries close to Poland (e.g. spatial slice predicate) before 1949 is not a valid query for GDR because it did not exist in 1949.

However, a more precise study of SOLAP exploration constraints is necessary to provide a comprehensive conceptual framework and effective implementation. Thus, in this paper, we present a new classification of SOLAP IC by detailing the exploration integrity constraint class introduced briefly in Boulil et al. (2012a, 2012b).

Furthermore, motivated by the relevance of defining ICs at a conceptual level and specifically using standard languages such as Unified Modelling Language (UML) and Object Constraint Language (OCL), in Boulil et al. (2012b) we proposed a UML profile and OCL statements for the specification of SOLAP IC. In this work, we present an extension of the UML profile aimed at the conceptual design of query (exploration) ICs.

Finally, motivated by the importance of an automatic implementation for avoiding errors related to the ad hoc application implementations of complex and iterative SOLAP applications, we propose a tool for the automatic implementation of SOLAP ICs in a relational SOLAP architecture. Finally, we validate our proposal in the context of a real case study concerning an energetic farm indicator analysis (Bimonte, Boulil, Pradel, & Chanut, 2012), which also represents an important contribution because the literature does not present real applications that validate the model-driven data warehouse development process, as also underlined in Benker and Jürck (2012).

The paper is organised as follows: Section 2 presents the main concepts of Spatial OLAP and the existing works on integrity constraints in (spatial) data warehouses; Section 3 introduces the case study and main concepts of our UML profile for the conceptual design of SOLAP models. In Section 4, our classification of SOLAP IC is detailed; Section 5 presents our framework for the conceptual specification of SOLAP IC; Section 6 describes our implementation; Section 7 presents a discussion of the limits of our proposal and highlights short- and long-term research perspectives.

2. Main concepts and related work

2.1. Spatial OLAP

Data warehouses and OLAP systems are widely recognised as decision support systems for the analysis of large volumes of alphanumeric data. These data are modelled using multidimensional models, which define the concepts of facts and dimensions (Kimball, 1996). Facts represent the subjects of analysis and are described by numerical measures, which are subsequently analysed at different granularities represented by the levels of hierarchies composing the dimensions. The business analyst explores the warehoused data (cube) through OLAP operators. The usual OLAP operators are: (1) slice, which selects a portion of the warehoused data; (2) dice, which projects a dimension; (3) roll-up, which aggregates measures using Structured Query Language (SQL) aggregate functions (sum, min, max, etc.) when climbing the dimension hierarchies; and (4) drill-down, which is the reverse of roll-up.

In SDW and SOLAP, the explicit representation of spatial data in dimension levels allows the visualisation of OLAP query results by means of maps and the use of topological, metrical and directional operators for slicing of multidimensional data (Bédard et al., 2006). The integration of spatial data as measures allows their aggregation using such functions as Centroid, Geometric union, etc., and therefore provides a ‘global’ and better understanding of the analysed phenomena.

A typical Spatial Relational OLAP (Spatial ROLAP) architecture is composed of three tiers (Figure 2): the SDW tier, the SOLAP server tier and the SOLAP client tier (Bimonte & Pinet, 2010).

Due to the important economic impact of erroneous data in BI systems (Eckerson, 2000), handling of quality issues in spatial and traditional data warehouses is a crucial research issue (Rizzi, Abelló, Lechtenböcker, & Trujillo, 2006). Integrity constraints represent an effective method to define quality rules for detecting inconsistent data, aggregations or queries in SDW systems (Boulil et al., 2012b). However, to ease the implementation process, which can be tedious, complex and iterative, conceptual models based on standard languages such as UML and OCL are mandatory for defining ICs at an early stage of the SDW design process as well as their automatic implementation. UML is widely accepted as the object-oriented standard for modelling of various aspects of software systems (OMG, 2011) in addition to SDW systems (Aguila, Fidalgo, & Mota, 2011). Indeed, any approach using UML minimises the efforts of designers and decision makers in developing and implementing the data schema. UML can also be interpreted by Computer-Aided Software Engineering (CASE) tools. In the same way, specifying ICs at a conceptual level allows the consideration of all possible quality problems and their validation by domain experts and users. This specification must be carried out using appropriate languages that are compatible with the data modelling language and that will ease the implementation of the constraints. In this context, OCL (OMG, 2006) represents an effective solution to define ICs at the conceptual level in a clear, non-ambiguous and platform-independent implementation effort in spatial databases (Pinet, Duboisset, & Soullignac, 2007) and SDW (Boulil et al., 2010). OCL is also integrated in CASE tools (such as Eclipse) and code generators, allowing automatic implementations of constraints. The automatic implementation improves the effectiveness of the design process by reducing errors and shortening the time of software development.

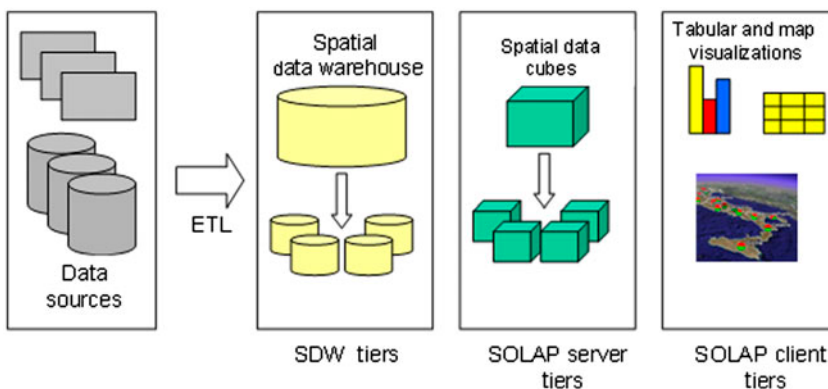


Figure 2. The relational SOLAP architecture.

Note: SOLAP: Spatial OnLine Analytical Processing; ETL: Extract, Transform and Load tools; SDW: Spatial Data Warehouse.

2.2. Spatial ICS

In the following, we describe the main works regarding the definition of ICs in (S)DW systems. In Carpani and Ruggia (2001) and Ghozzi, Ravat, Teste, and Zurfluh (2003), authors propose some ICs for classical data warehouses (for example, an inter-member IC expresses a constraint between two or more dimension members; (Carpani & Ruggia, 2001)) specified using logical predicates on top of ad hoc conceptual models. Despite the importance of these preliminary works, they do not consider all possible types of SOLAP IC (e.g. they do not support IC on facts), and they are based on non-standard conceptual models and finally do not provide any implementation. Malinowski and Zimányi (2008) propose an extension of the ER model for the design of complex spatio-temporal data warehouses. This model is enriched with a set of pictograms to express certain spatial Data ICs, namely, the ICs representing the allowed topological relationships (e.g. coveredBy) between dimension members. The implementation is demonstrated by detailing some mapping rules between the conceptual model and Oracle Spatial. Despite the relevance of this work, it covers only some classes of ICs and does not propose any architecture for automatic implementation. Glorio and Trujillo (2008) propose a UML profile for SDW conceptual design, but consider a small number of Data ICs. The authors express only domain values and multiplicity constraints of attributes and multidimensional relationships (levels and fact-dimension relationships). A survey on aggregation issues is presented in Mazón, Lechtenbörger, and Trujillo (2009). This work expresses simple structural constraints with UML multiplicities (e.g. facts should be linked to dimensions with one-to-many relationships) to ensure correct aggregations. However, this work does not propose any implementation for the specified ICs.

To best of our knowledge only Bimonte, Villanova-Oliver, and Gensel (2010) and Pedersen and Tryfona (2001) address problems related to aggregation of measures in spatial data warehouses. In particular, Bimonte et al. (2010) investigates the relations between aggregation of spatial and numerical measures, and Pedersen and Tryfona (2001) provides a framework to avoid summarisability problems related to spatial members that are not topologically disjoint. However, these works do not adopt a standard representation or provide an automatic implementation.

In Pinet and Schneider (2010), complex structural aggregation and data constraints are expressed using UML constructs and OCL. As with other related works, Pinet and Schneider (2010) do not allow presentation of semantic aggregation constraints and they do not provide an automatic implementation. Boulil et al. (2010) presents the specification (on top of a UML-based SDW conceptual model) of many Data ICs on warehoused spatial data by means of Spatial OCL, which is an extension of OCL that allows the expression of topological relationships between simple and complex geometries defined in Pinet et al. (2007). This work also proposes an automatic implementation using a code generator (Spatial OCL2SQL) in the Spatial DBMS Oracle Spatial 11 g. Finally, Levesque et al. (2007) proposes an approach for preventing the risks of data misuse in SOLAP systems. This approach is adapted from existing methods of risk management and consists of classical steps such as risk identification, analysis and documentation, among others. Specifically, the risk documentation is carried out using paper forms. Because this work is intended for data producers, it proposes neither a specification language for these quality problems nor implementation techniques to be used by SDW designers.

As shown in the detailed comparison presented in Boulil (2012), to the best of our knowledge, no previous work proposes a standard-based specification and an automatic implementation of the three SOLAP IC types.

3. Preliminaries

In this section, we present the main concepts of our UML profile for the conceptual design of spatial data cubes proposed in Boulil (2012) that we extend to express SOLAP ICs at the conceptual level (Section 3.1). Furthermore, the case study applied in the paper illustrates and validates our proposal (Section 3.2).

3.1. UML Spatial data cube profile

When an extension of UML is required, it is possible to directly modify the different elements, e.g. classes or relationships, of the UML metamodel (changing attributes, adding new relationships, etc.), but this technique is usually not implemented in UML-based tools. By contrast, the concept of the UML profile is the standard technique for the customization of UML for particular domains or platforms. Indeed, UML profiles allow extended metaclasses (class, property, etc.) (OMG, 2011). A profile is defined using three extension mechanisms: stereotypes, tagged values and constraints. A stereotype is an extension of a UML metaclass, and it is represented using the notation <<stereotype-name>> and/or an icon. For example, it is possible to create a stereotype <<GeographicClass>> that extends the UML metaclass 'class'. The tagged values are meta-attributes; they are defined as the properties of the stereotypes. Finally, a set of constraints should be attached to each stereotype to precisely define the application semantics and to avoid arbitrary use by the designers (e.g. a constraint can be defined to guarantee that a <<GeographicClass>> class has a geometric attribute known as 'geometry'). These constraints are often defined using OCL, which has been adopted by the Object Management Group as the standard language for expressing constraints in UML models. OCL provides a platform-independent and generic method for modelling constraints, and OCL can be interpreted by code generators to produce code automatically. In fact, certain tools allow the production of integrity-checking mechanisms in different languages (Java, C#, SQL, etc.) from the specifications of the constraints expressed in OCL. The OCL constraints can be applied at the meta-model level (e.g. UML profile) and also at the model level (the instance of the UML profile). Spatial OCL is an extension of OCL that supports topological spatial relationships (inside, intersect, etc.) (Pinet et al., 2007).

An example of a Spatial OCL constraint stating that the geometries of the instances of the class 'City', designed using the stereotype <<GeographicClass>> must be topologically disjoint is:

```
context City inv:
  City.allInstances->forAll(c1, c2| c1<>c2 implies
    c1.geometry.disjoint (c2.geometry))
```

In Boulil et al. (2012a), we presented a UML profile for the conceptual design of spatial data cubes. The profile is organised in two parts: one part describes the SDW multidimensional structures (*SDW metamodel*) and the second represents how measures are aggregated with respect to the analysis needs of the decision makers (*Aggregation metamodel*). The SDW metamodel allows conceptual representation of classical and advanced aspects of the spatio-multidimensional model, including multiple and complex hierarchies and many-to-many relationships between facts and dimensions, among others. The profile defines a stereotype or a tagged value for each spatio-multidimensional element; for example (see Figure 3), 'Fact' for the fact, 'SpatialAggLevel' for the

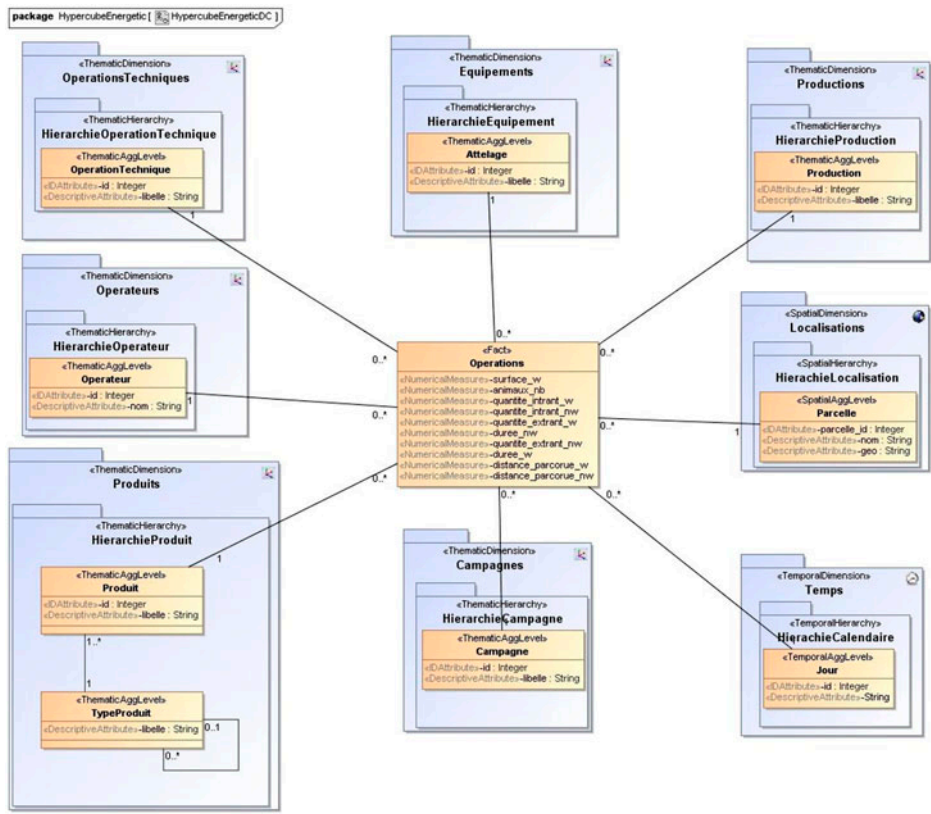


Figure 3. The SDW Model of the Energetic SOLAP application.
Note: SDW: Spatial Data Warehouse; SOLAP: Spatial OnLine Analytical Processing.

spatial dimension levels whose geometries are represented with geometric attributes stereotyped ‘LevelGeometry’ and ‘AggRelationship’ to model the aggregation relationships that link dimension levels.

The Aggregation metamodel is used to represent the way in which measures are aggregated along dimension hierarchies. In particular, the stereotype ‘BasicIndicator’, which extends the UML metaclass ‘class’, allows representation of the analysis indicators (e.g. ‘Sum_Quantity_Intrant_W’ of Figure 4); it defines a set of aggregation rules over a given measure, which is specified by the ‘aggregatedAttribute’ property. These aggregation rules define the functions used to aggregate measures along dimensions

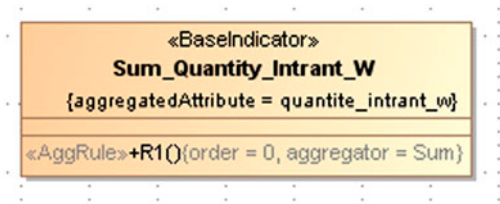


Figure 4. An excerpt of the Energetic Aggregation Model.

and hierarchies, and they are specific UML operations, which are specified using different stereotypes, such as the ‘AggRule’ stereotype (Figure 4).

3.2. Case study

Recently, public and private organisations have recognised the widely important impact of energy consumption on environmental, social and economic realities. In particular, several actions are in process for reduction of agricultural energy consumption to improve economic development and decrease its environmental impact. In the context of a French project, we have developed a SOLAP application for the analysis of energetic farm indicators. The spatio-multidimensional model of this application is shown in Figure 3 and is organised around the analysis of the consumed energy (using measures such as ‘quantite_intrant_w’, ‘quantite_extrant_w’, etc.), the worked area and the treated animals for a particular operation (‘Opération technique’). Additional details can be found in Bimonte et al. (2012).

These measures are analysed along seven dimensions:

- Campaigns (‘Campagnes’): production cycles expressed in terms of years;
- Time (‘Temps’): a classical temporal dimension;
- Products (‘Produits’): the input and output products of farming operations;
- Operators (‘Opérateurs’): persons who perform the operations;
- Equipment (‘Equipements’): machines and tools used during the operations;
- Location (‘Localisations’): a spatial dimension that groups plots (‘Parcelles’) where the operations take place by farm (‘Exploitation’), farms by department (‘Département’) and departments by region (‘Région’);
- Productions: representing the type of production (e.g. wheat);
- Technical Operations (‘OperationsTechniques’): the performed technical operations, which are grouped by functions.

An excerpt of the aggregation model of this application is depicted in Figure 4, and represents an analysis indicator, ‘Sum_Quantity_Intrant_W’, which is defined by summing the measure ‘quantite_intrant_w’ along all dimensions. In this case study, all measures are aggregated using the sum function but, if necessary, designers can specify more complex indicators using the different types of aggregate functions and aggregation rules defined in our profile.

Finally, using this model, users can answer OLAP queries such as ‘What is the total quantity (‘Sum_Quantity_Intrant_W’) of the consumed fuel for a spreading operation on 19 September 2000 in the Montoldre farm?’ or ‘What are the surfaces of the worked areas per plot and operations in 2000?’

4. Solap IC classification

In this section, we present an extension of our previous SOLAP IC classification proposed in Boulil et al. (2012a, 2012b) by specialising and detailing the SOLAP exploration ICs (Query IC subclasses). This classification (Figure 5) serves as a reference guide for the process of handling the three types of quality control issues in a SOLAP architecture.

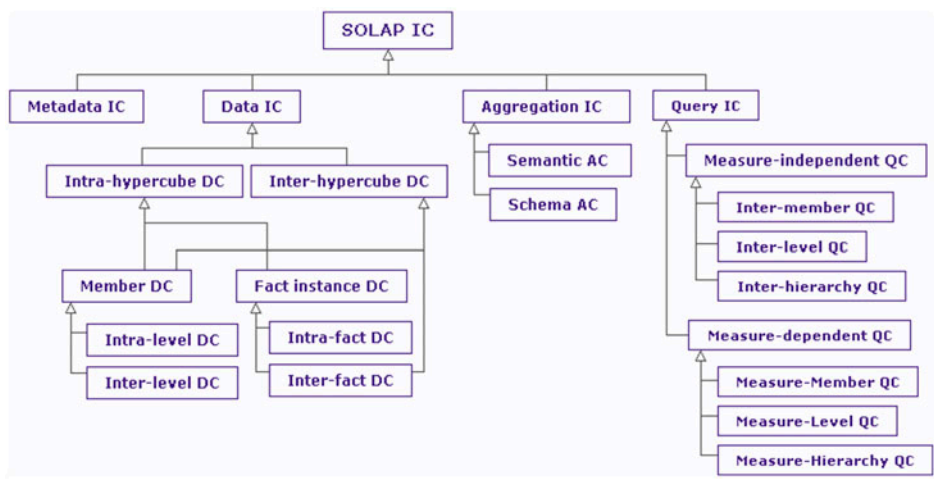


Figure 5. SOLAP IC classification.
Note: IC: Integrity Constraint; SOLAP: Spatial OnLine Analytical Processing.

We provide explanations and selected examples of these IC classes using the previously described case study.

As shown in Boulil et al. (2011), *Metadata IC* verifies the consistency of metadata from different integrated data sources (e.g. spatial members and measures must be defined with the same geographic scale). This type of constraints can be defined on all categories of metadata (e.g. metadata on facts, dimensions, SOLAP visualisation, etc.), but they have not been investigated in this work.

Data IC usually ensures the semantic consistency of warehoused (spatial) data (Pinet, Duboisset, & Schneider, 2011). The semantic consistency is related to distance between the data and the real world. As indicated in Pinet et al. (2011), *Data IC* provides a framework for adding semantics to models; ICs are properties which should be satisfied by all instances of a data schema.

Example 1. ‘The geometry of each parcel must be spatially included into the geometry of its farm’.

These constraints can be defined on all elements of the SDW, including facts, members, etc.

Aggregation IC guarantees the correct and meaningful aggregations of measures. In particular, *Semantic Aggregation IC* addresses the problem of the applicability of aggregate functions to measures according to the semantic natures and the types of measures, aggregate functions and dimensions. For example: ‘The sum of temperature values does not make sense’.

Example 2. ‘It is not possible to aggregate the number of the values of the measure “duree-w”, which is a numeric measure representing the durations of farming activities, using the aggregate function Geometric Union’.

Schema Aggregation IC includes conditions that must be satisfied by dimension hierarchies and dimension-fact relationships to avoid double counting and incomplete

aggregates. For example, dimensions and facts should be associated by one-to-many relationships (Mazón et al., 2009).

Query IC refers to conditions that guarantee that user queries formulated by selecting and combining data-cube elements in the SOLAP client are consistent to avoid problems of interpretation. To ensure the consistency of the SOLAP query, two aspects should be controlled: (1) its structural validity (i.e. if the combination of data-cube elements defined by the user forms a structurally valid multidimensional query) and (2) its semantic validity and satisfiability (i.e. there exist meaningful results for the query) with respect to the business rules of the application domain. The structural validity is handled by the main (S)OLAP servers (e.g. Mondrian, Microsoft AS) and does not require further consideration. Consequently, the Query IC we define in this work deal only with the semantic aspects.

For example, the (S)OLAP query ‘What were the average temperatures in the USSR in 2010?’ (for the sake of simplicity we do not use spatial predicates in this example) returns an empty result because no temperature values exist for the USSR after 26 December 1991 (the dissolution date of the USSR). Even if this IC is implemented as a Data IC, classical SOLAP tools allow decision makers to formulate this query by combining these two members (USSR and 2010) and returning an empty value. This situation leads to a problem of interpretation: this empty value may be perceived as if there were no temperature values registered for the USSR during 2010 instead of as the realisation that this combination of members (USSR and 2010) is impossible. Although this query example could be resolved by use of particular spatio-multidimensional data structures, such as DW versioning structures (Arigon, Miquel, & Tchounikine, 2007), the Query IC allows designers to model any invalid query which may be independent of time-versioning aspects (for example, certain products cannot be sold in certain stores).

In particular we have identified six classes of SOLAP Query ICs that consider the involved spatial data-cube elements. These IC classes are grouped into two classes: (1) measure-independent and (2) measure-dependent Query ICs.

(1) *Measure-independent Query IC* defines invalid combinations of dimensional elements independently of the measures and indicators (i.e. these combinations are invalid regardless of the selected measure) and are classified into three sub-classes:

Inter-member Query IC prevents invalid combinations of two or more dimension members. An example is the previous query combining the USSR with 2010.

Inter-level Query IC prohibits any combination of members from two or more dimension levels. For example, ‘it is not possible to combine French departments (“Department” level) with US products (“US Product” level)’.

Inter-hierarchy Query IC forbids any combination of members from two or more dimension hierarchies. For example, under certain business rules, combining French products (members of the French products hierarchy) with US stores (members of the spatial dimension hierarchy representing US stores) can be incorrect.

(2) *Measure-dependent Query IC*. None of the previously described query constraints addresses measures, i.e. they are valid for all measures of the SOLAP model. However, it is possible to encounter more complex situations in which these constraints are pertinent only for a subset of measures. We refer to these integrity constraints as *Measure-dependent Query IC* constraints. These constraints are also subdivided into three sub-classes:

Measure-Member Query IC forbids the combination of a subset of measures with a member set or a combination of member sets.

Example 3. ‘It is not possible to obtain a value for the measure representing the number of animals for the spreading technical operation’.

Measure-Level Query IC forbids the combination of a subset of measures with a dimension level or a combination of dimension levels.

Measure-Hierarchy Query IC forbids the combination of a subset of measures with a hierarchy or a combination of hierarchies.

5. The conceptual framework

To define SOLAP data, aggregation and exploration ICs at a conceptual level, we propose a framework based on a UML profile and Spatial OCL (Figure 6). The main aim is to build a unique UML profile that defines three interconnected metamodels that conceptually represent: (1) SDW data structures and their Data ICs (SDW metamodel), (2) how measures are aggregated to meet the analysis requirements and aggregation ICs (Aggregation metamodel) and (3) the Query IC (Query IC metamodel).

In particular, Data ICs are defined by designers using Spatial OCL on top of the SDW model, the Aggregation ICs are defined as the Aggregation metamodel’s stereotype constraints using OCL and the Query ICs are defined using the Query IC metamodel and Spatial OCL.

5.1. Data and Aggregation IC

The SDW metamodel allows the definition of SDW data structures and the expression of Data ICs on top of these structures using Spatial OCL (Boulil et al., 2010). The Data IC of Example 1 (‘the geometry of each parcel must be spatially included in the geometry of its farm’) is expressed using Spatial OCL as follows:

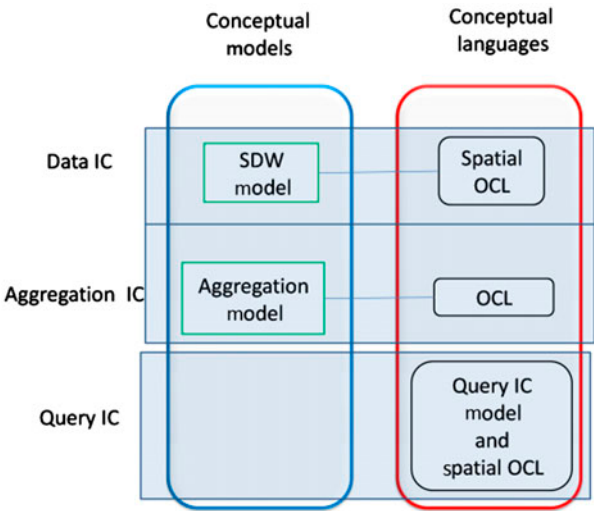


Figure 6. The UML/OCL-based conceptual framework for SOLAP ICs.
Note: IC: Integrity Constraint; OCL: Object Constraint Language; UML: Unified Modeling Language; SDW: Spatial Data Warehouse; SOLAP: Spatial OnLine Analytical Processing.

```
context Parcelle inv DataIC1:
self.geo.isInside (self.exploitation.geo) or
self.geo.coveredBy (self.exploitation.geo)
```

In Boulil et al. (2011), we identified a set of semantic aggregation constraints that ensure meaningful aggregation of measures. These constraints are valid for all SOLAP applications. Thus, we have implemented these constraints as OCL constraints in the Aggregation Model package of the profile. The constraints are checked by the CASE tool at the design stage when validating the conceptual model. For example, a semantic Aggregation IC that checks whether the aggregate function type conforms to (i.e. is applicable to) the measure data type (Example 2) is defined by the following OCL statement:

```
context AggRule inv AggregatorAndMeasureTypeConformity:
not (baseIndicator.aggregatedAttribute.OclIsUndefined ()) implies
agggregator.applicableTo->exists (dt|
baseIndicator.aggregatedAttribute.oclsTypeOf (dt))
```

This OCL constraint verifies that the measure's data are included in the set of data types to which the aggregator can apply.

We will consider semantic aggregation constraints defined in Bimonte et al. (2010) and Pedersen and Tryfona (2001) in future work.

5.2 Query IC

To express Query ICs, we use the UML profile presented in Figure 7 and Spatial OCL. A Query IC (stereotype 'SDWQueryConstraint') is formalised as a specialisation of the UML metaclass 'class' and specialised in 'HierarchyQC', 'LevelQC' and 'MemberQC' representing the Query IC classes previously described in Section 3. The 'HierarchyQC' stereotype allows the specification of constraints of the Inter-hierarchy QC and Measure-Hierarchy QC classes (Section 3) and contains the property 'onHierarchies' indicating the hierarchies (Hierarchy) involved in the IC; these hierarchies are defined in the SDW model. In the same way, the 'LevelQC' stereotype allows specification of the constraints of Inter-level QC and Measure-Level QC classes (described in Section 3); the involved dimension levels ('AggLevel' stereotype) are defined using the 'onLevels' property. Finally, the 'MemberQC' stereotype represents the constraints of the Inter-member QC and Measure-Member QC classes; the sub-sets of members involved in the constraint are represented as UML attributes and formalised by the 'MemberSet' stereotype. Each 'MemberSet' attribute selects a sub-set of members of a dimension level ('MemberSet' stereotype) by defining a condition of selection over its (the dimension level) members. This condition is defined as an OCL constraint ('MemberSelection' stereotype).

The dependence or independence from measures of the query constraint ('SDW-QueryConstraint') is specified using the 'onMeasures' property (multiplicity 0..*, where 0 indicates that no measure is involved).

To precisely define the semantics of each stereotype and avoid invalid specifications of constraints by designers using this component of our profile (Figure 7), we have defined a set of OCL constraints. For example, the following constraint states that if the 'HierarchyQC' constraint is independent from measures (i.e. the set of involved measures 'onMeasures' is empty), then at least two hierarchies must be involved in the constraint; otherwise, at least one hierarchy is necessary.

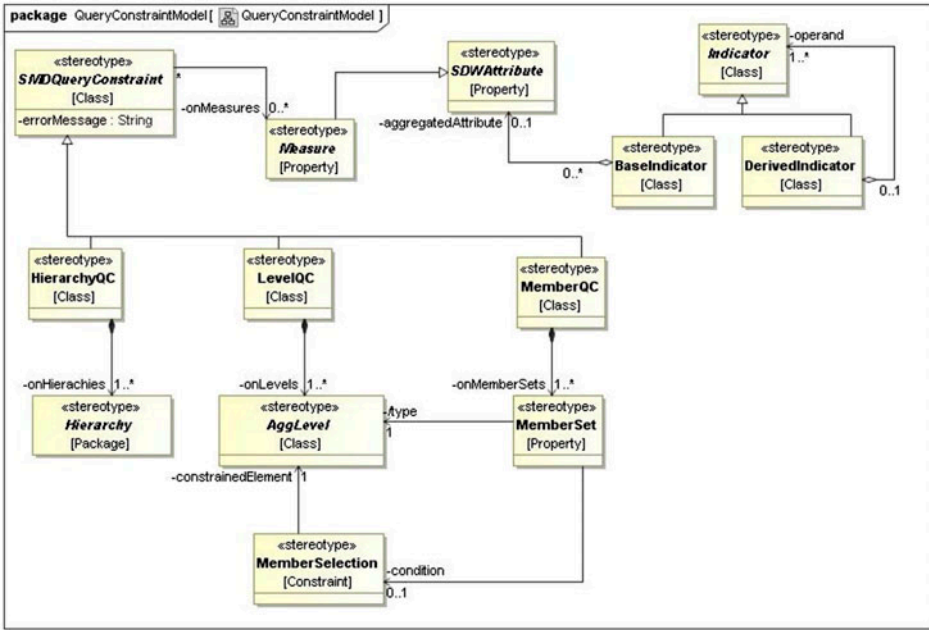


Figure 7. Query IC Metamodel.

Note: IC: Integrity Constraint

```

context HierarchyQC inv atLeastOneOrTwoHierarchies:
if (self.onMeasures->isEmpty ())
then (self.onHierarchies->size () >= 2)
else (self.onHierarchies->size () >= 1)
endif

```

This constraint is defined in the context of the *HierarchyQC* stereotype. Similar constraints are defined on the ‘*LevelQC*’ and ‘*MemberQC*’ stereotypes. As described in Section 3, the formal definition of the UML profile constraints in OCL allows Magic-Draw (the CASE tool) to check whether the SOLAP model specified by the designer is well defined. For example, if the designer defines a model that does not respect the previously described OCL statement, an error message is displayed to him/her indicating that the model is not valid, as shown in Figure 8; this is an example of a *HierarchyQC* constraint that involves one hierarchy (‘*HierarchyProduit*’) and no measures.

An example of a Measure-Member Query IC is depicted in Figure 9, and states that combining the technical operation (<<MemberSet>> ‘*OperationTechnique*’) ‘spreading’ (condition = *operationTechniqueEqualsEpannage*, whose OCL expression is shown in Figure 10) with the measure ‘number of animals’ (*onMeasures* = *animaux_nb*) is meaningless in any SOLAP query.

6. Implementation

In this section, we present the architecture designed to automatically implement the SOLAP ICs. The main aim is to automatically implement each type of IC in a different tier of a classical SOLAP architecture (Figure 1). The conceptual definition of each IC is automatically translated into the implementation language supported by the SOLAP

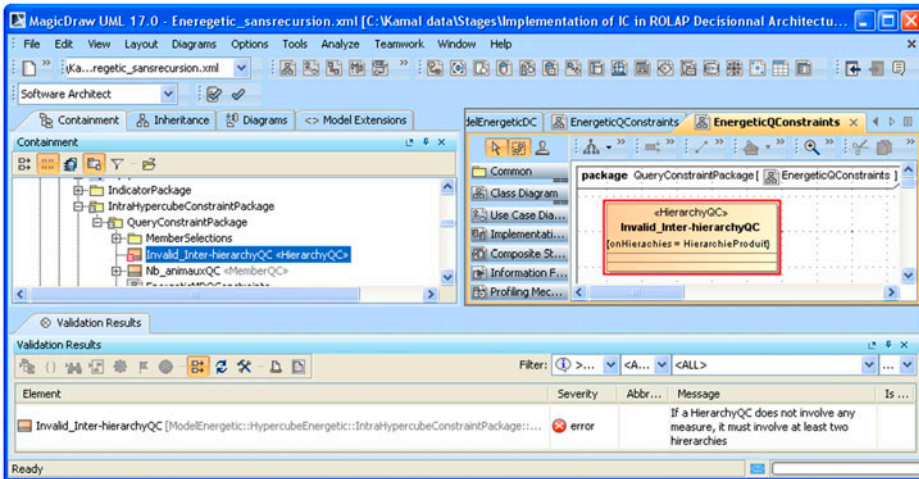


Figure 8. An example of a verification of a Query IC profile with an inherent constraint.
Note: IC: Integrity Constraint.

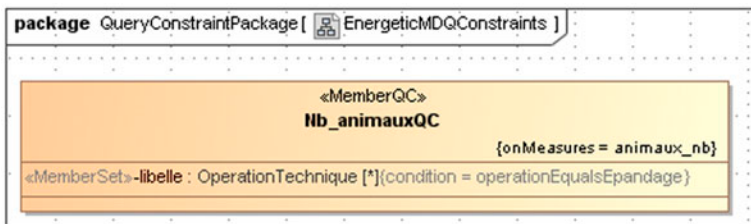


Figure 9. An example of a specification of a Query IC using our profile.
Note: IC: Integrity Constraint.

```
context OperationTechnique inv :
    self.libelle = 'Epannage'
```

Figure 10. OCL statement of MemberSet of Figure 9.
Note: OCL: Object Constraint Language.

tier in which it is appropriate to implement the constraint. In particular, Data ICs are translated using SpatialOCL2SQL and implemented in the SDW tier, Query ICs are translated by our automatic code generator (known as UML2MDX) and implemented in the OLAP server and, finally, Aggregation ICs are integrated into our profile definition, which is implemented as a module of the MagicDraw CASE tool. These constraints are controlled during the design stage by the CASE tool.

Our SOLAP architecture (Figure 11) is based on Spatial DBMS Oracle Spatial 11 g (Oracle, 2012), the OLAP Server Mondrian (Pentaho, 2012) and the SOLAP client JRubik (JRubik, 2012). Oracle Spatial 11 g is a relational DBMS that provides native support for storing and querying of spatial data, and Mondrian is a software package designed to provide OLAP functionality in an open and extensible framework on top of a relational database. Mondrian includes a Calculation layer that validates and executes Multidimensional Expressions (MDX) (Microsoft, 2012) queries and an

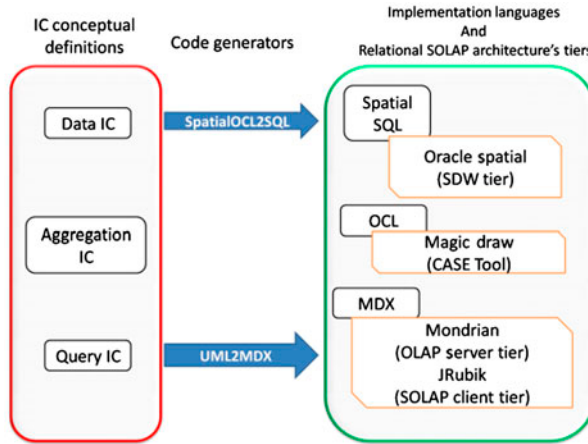


Figure 11. The framework for an automated implementation of SOLAP ICs.

Note: IC: Integrity Constraint; MDX: Multidimensional Expressions Language; OCL: Object Constraint Language; OLAP: OnLine Analytical Processing; SOLAP: Spatial OnLine Analytical Processing; SQL: Structured Query Language; SDW: Spatial Data Warehouse.

Aggregation layer that controls data in memory and requests data that are not cached. MDX is a standard language used to query multidimensional databases, similar to SQL for relational databases. To guarantee the greatest flexibility and to interface with the relational data, an eXtensible Markup Language (XML) description of the multidimensional application is used. JRubik is a software package designed to provide a graphical presentation layer on top of Mondrian and provides the functionality for modification of the visualisation of the pivot table and triggers the desired OLAP operators of drill-down replace, drill-down position, expand-all and drill-through. JRubik also allows the cartographic representation of SOLAP queries.

A demo of the proposed implementation is available at (Nazih et al., 2012).

6.1. Data IC

In our approach, designers specify Data IC with the text editor of Spatial OCL2SQL (Pinet et al., 2007). The code generator of Spatial OCL2SQL will produce automatically a SQL code corresponding to the Data IC. This SQL code will be used in the DW to forbid the insertion of inconsistent tuples. In the case of an already loaded DW, Spatial OCL2SQL also produces SQL queries that can be used to select data that do not satisfy the IC conditions in the DW. This technique allows designers to easily identify inconsistent data in the DW. Note also that checking the constraints at each tuple insertion (during the ETL process) is less efficient in term of time than checking all the constraints after the loading of all the data in the DW.

More precisely, to automatically implement the Data ICs in Oracle 11 g, we use the code generator Spatial OCL2SQL (Figure 11). Spatial OCL2SQL is a Java-based open source tool that integrates the spatial extensions of OCL known as OCL 9IM and OCL ADV (Pinet et al., 2007) and automatically generates SQL scripts for Oracle from the Spatial OCL constraints. Its main inputs are: (1) a conceptual data model defined by an XML Metadata Interchange (XMI) schema, (2) an OCL constraint file and (3) a geometric metadata file. The tool outputs SQL scripts to create the SDW physical

schema in the Oracle DBMS and a set of SQL queries and triggers to implement the OCL constraints for data integrity checks.

More precisely, this tool transforms each Spatial OCL constraint into a SQL query, a SQL view and a set of triggers. The query and the view select the data (n-tuples of the table corresponding to the contextual class of the Spatial OCL IC) that do not satisfy the constraint. For example, the SQL view generated for the previously defined OCL Data IC of **example 1** is the following:

```
create or replace view DataIC1View as
(select * from PARCELLE SELF where not (
(MDSYS.SDO_RELATE (
(select GEO from EXPLOITATION where PK16 in (select
RELATED_EXPLOITATION_PK16 from OV_PARCELLE where PK5 = SELF.PK5)),
SELF.GEO,
'mask=CONTAINS querytype=WINDOW') = 'TRUE') OR
(MDSYS.SDO_RELATE(
(select GEO from EXPLOITATION where PK16 in (select
RELATED_EXPLOITATION_PK16 from OV_PARCELLE where PK5 = SELF.PK5)),
SELF.GEO,
'mask=COVERS querytype=WINDOW') = 'TRUE')
));
```

This SQL view selects the n-tuples of the table PARCELLE that do not satisfy the Data IC condition. The triggers (one trigger for each table involved in the IC) can be used to verify whether the SQL view is non-empty when updating the corresponding table, in which case an error message is returned. For example, the following trigger defined on the table EXPLOITATION verifies whether the SQL view DataIC1View is non-empty when updating the table EXPLOITATION:

```
create or replace trigger tr_DataIC1View_on_EXPLOITATION
after insert or update or delete on EXPLOITATION
declare tmp number;
begin
select NVL(COUNT(*), 0) into tmp from DataIC1View;
if (tmp > 0) then
raise_application_error (-20,000, 'integrity violation at DataIC1View');
end if;
end;
```

6.2. Aggregation IC

MagicDraw (No Magic, 2012) supports OCL at the meta-model (UML profile) level. In other words, MagicDraw is able to check the OCL constraints defined on UML stereotypes at the conceptual design stage (Figure 4). This functionality allows checking of the Aggregation IC independently of the specific SOLAP architecture and without providing any implementation efforts. For example, if the designer defines an analysis indicator using the GeometricUnion to aggregate values of the numeric measure 'duree_w' to represent the durations of farming operations (Example 2), then MagicDraw, by checking the OCL semantic aggregation constraint previously defined in Section 4.1, informs the designer that the constraint is not satisfied, as shown in Figure 12.

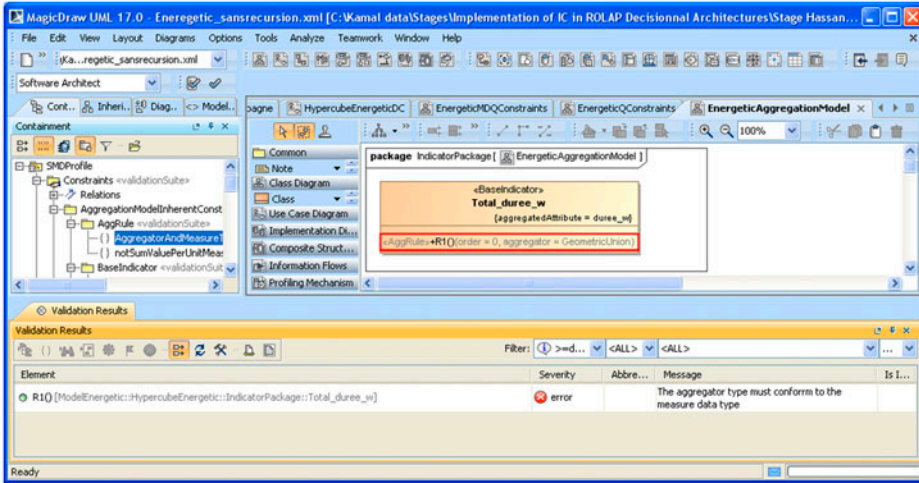


Figure 12. An example verification of an aggregation IC.

6.3. Query IC

The automatic implementation of Query IC defined by the users with MagicDraw is based on MDX, which is a standard for OLAP Servers/Clients (Figure 4). Thus, the choice of Mondrian as an OLAP server is not a limitation for our generic architecture because our MDX implementation is platform-independent. The main aim is to translate the Query IC into an MDX formula, which is stored in the OLAP Server and subsequently visualised in the SOLAP client. These formulas, when executed, inform the user of incorrect results. For each Query IC type, we have defined an MDX template. The templates are fulfilled using a Java method (UML2MDX) that parses the XMI files associated with the Query IC.

The MDX code implementing the Query IC of Example 3 (Figures 9 and 10) is shown in Figure 13, and an example of its verification on an OLAP query is shown in Figure 14. We note that different visual policies are applied to the cells of the JRubik pivot table, which are defined by different combinations of the indicator ‘Max_animaux_nb’ and members of the dimension ‘Technical Operations’; the invalid cells that involve the member spreading (‘Epannage’) are displayed with a red colour, the valid cells (such as the one involving the member milking operation ‘Traire’) are displayed with a green colour and, finally, the aggregated cells composed of valid and invalid cells (such as the cell involving the member ‘AllOperationsTechniques’) are represented with a yellow colour.

This MDX implementation, which consists of a calculated member defined on the involved measure ‘animaux-nb’, defines two nested tests using the MDX function *IIF* (condition, action 1 if true, action 2 if false). First (the first *IIF expression*), it verifies whether the selected member from the dimension ‘OperationTechniques’ is ‘Epannage’ or a descendant of ‘Epannage’ in which case it depicts the cell with a red colour. If not, it verifies (the second *IIF expression*) whether the selected member is an ascendant of ‘Epannage’, in which case the cell is formatted with a yellow colour; otherwise it is depicted with green colour.

To calculate the ‘OperationTechniques’ members whose name is ‘Epan dage’ as well as their descendants, and ascendants, we use the following MDX operators: FILTER (returns a subset of dimension members that satisfy a given condition), DESCENDANTS (returns the descendants of a member at a given level) and ANCESTOR (returns the ancestor of a member at a given level).

7. Discussion and future work

The SDW and SOLAP systems represent effective decision-making technologies for the analysis of large geo-referenced data sets. The correctness of the SOLAP analysis depends on the warehoused data quality, how measures are aggregated to compute the data cubes and how the data cubes are explored. It is therefore necessary to control all of these aspects. To achieve this goal, certain works have proposed the definition of Integrity Constraints but only for the purpose of ensuring consistent data and aggregations.

Therefore, we have extended the notion of ICs in SOLAP systems to further control the consistency of user queries. Moreover, as shown in this paper, there is a lack of proposals that provide DW designers with effective solutions for standard-based specifications and automated implementation of SOLAP ICs. Subsequently, in this paper, we extend our framework based on standard languages (UML, Spatial OCL) and code generators (Boulil et al., 2012a, 2012b) to specify and automatically implement SOLAP Query ICs.

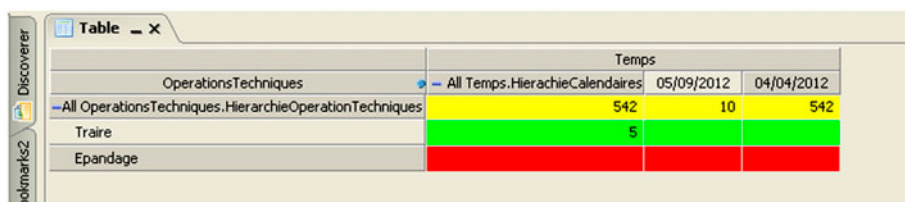
However, to consider all SOLAP ICs, certain issues should be resolved. In particular, to express additional SOLAP ICs with Spatial OCL, it is mandatory to extend this language with spatial metrical and ordinal relations (Parent, Spaccapietra, & Zimányi, 1999) because this approach only supports the 9-intersection model’s spatial topological relations between simple and complex geometries (Parent, Spaccapietra, & Zimányi, 1999). The integration of spatial metrical relationships, for example, can be useful for expressing some Data ICs, such as ‘the parcel areas (calculated using the metric operator “area”) should be greater than a certain value’. The extension of OCL with these spatial operators could be accomplished in a manner similar to the topological spatial relationships (Duboisset, Pinet, Kang, & Schneider, 2007).

Another important issue involves the integration of aggregate functions. OCL supports only the sum function and, in SOLAP applications, other aggregate, numerical and spatial functions may be necessary to express for example a Data IC stating that

```
<!-- MDX Implementation of Query IC -->

<CalculatedMember name="Max Nb Animaux_QC" dimension="Measures">
<formula> [Measures].[Max Nb Animaux] </formula>
<CalculatedMemberProperty name="FORMAT_STRING" expression=" IIF((([OperationsTechniques].CurrentMember in
Filter([OperationsTechniques].Members,[OperationsTechniques].CurrentMember.Name='Epandage')) or
([OperationsTechniques].CurrentMember.Level.Ordinal > [OperationsTechniques].[OperationTechnique].Ordinal) AND
(Anccestor([OperationsTechniques].CurrentMember,[OperationsTechniques].[OperationTechnique]) in
Filter([OperationsTechniques].[OperationTechnique].Members,
[OperationsTechniques].[OperationTechnique].CurrentMember.Name='Epandage')
))),
&quot;;|#|style=red&quot;;
IIF([OperationsTechniques].CurrentMember in Filter([OperationsTechniques].Members,
Intersect(Descendants([OperationsTechniques].CurrentMember,[OperationsTechniques].[OperationTechnique].SELF),
Filter([OperationsTechniques].[OperationTechnique].Members,
[OperationsTechniques].[OperationTechnique].CurrentMember.Name='Epandage'))).COUNT>0),
&quot;;|#|style= yellow&quot;;
&quot;;|#|style=green&quot;; )" />
</CalculatedMember>
```

Figure 13. The MDX code implementing the Query IC of Example 3.
Note: IC: Integrity Constraint; MDX: Multidimensional Expressions Language.



OperationsTechniques	Temps	
	05/09/2012	04/04/2012
All OperationsTechniques.HierarchieOperationTechniques	542	10
Traire	5	
Epannage		

Figure 14. An example of a verification of the Query IC of Example 3.

Note: IC: Integrity Constraint

the spatial union of all plot geometries must be equal to the geometry of their farms. This point could be achieved in the same way as in Cabot, Mazón, Pardillo, and Trujillo (2010), in which the authors use OCL types to define a set of such numerical aggregate functions as Average, Standard Deviation, etc.

Moreover, other IC classes could be considered, including metadata and domain-specific aggregation ICs. For example, to express metadata constraints, our profile can be easily extended with tagged values and stereotypes to represent the main types of metadata, including scale, spatial reference, measure unit, etc. (ISO, 2003). Subsequently, OCL can be used to express certain constraints on these elements in a manner similar to that of the semantic aggregation ICs.

A formalisation of the proposed IC classification, consistent with that of Salehi, Bédard, Mostafavi, and Brodeur (2011), should be necessary to validate the exhaustivity, the expressivity and the completeness of our theoretical approach. Finally, it is highly important to include automated implementations of SOLAP ICs in different OLAP architecture types (Multidimensional OnLine Analytical Processing (MOLAP) and Hybrid OnLine Analytical Processing (HOLAP)) and technologies for the definition of an architecture for SOLAP ICs following the Model Driven Architecture (MDA) principles (OMG, 2003).

Metrics to measure data, control and analysis quality could also be considered in order to evaluate the gain in quality obtained by our approach.

References

- Aguila, P. S. R. D., Fidalgo, R. d. N., & Mota, A. (2011). *Towards a more straightforward and more expressive metamodel for SDW modeling* (pp. 31–36). Proceedings of the ACM 14th International workshop on Data Warehousing and OLAP, ACM, Glasgow, Scotland, UK.
- Arigon, A.-M., Miquel, M., & Tchounikine, A. (2007). Multimedia data warehouses: A multiversion model and a medical application. *Multimedia Tools and Applications*, 35, 91–108.
- Bédard, Y., Rivest, S., & Proulx, M.-J. (2006). Spatial on-line analytical processing (SOLAP): Concepts, architectures, and solutions from a geomatics engineering perspective. In R. Wrembel & C. Koncilia (Eds.), *Data warehouses and OLAP: Concepts, architecture, and solutions* (pp. 298–319). Idea Group Publishing.
- Benker, B., & Jürck, C. (2012). A Case study on model-driven data warehouse development. *DaWaK*, 54–64.
- Bimonte, S., Boulil, K., Pradel, M., & Chanet, J.-P. (2012). *Definition and analysis of new agricultural farm energetic indicators using spatial OLAP* (pp. 373–385). 17th International Conference on Geographical Analysis, Urban Modeling, Spatial Statistics (GEOG-AN-MOD 2012), Salvador de Bahia, Brazil.
- Bimonte, S., & Pinet, F. (2010). When Spatial analysis meets OLAP: Multidimensional model and operators. *International Journal of Data Warehousing and Mining*, 6, 33–60.

- Bimonte, S., Villanova-Oliver, M., & Gensel, J. (2010). *A multidimensional model for correct aggregation of geographic measures. Evolving application domains of data warehousing and mining: Trends and solutions* (pp. 162–183). IGI Global.
- 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-environnemental* (PhD thesis). Université Blaise Pascal, Clermont-Ferrand, France.
- Boulil, K., Bimonte, S., Mahboubi, H., & Pinet, F. (2010). *Towards the definition of spatial data warehouses integrity constraints with spatial OCL* (pp. 31–36). Proceedings of the ACM 13th international workshop on Data warehousing and OLAP, ACM, Toronto, ON, Canada.
- Boulil, K., Bimonte, S., & Pinet, F. (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, 11–39.
- Boulil, K., Bimonte, S., & Pinet, F. (2012a). *A UML & Spatial OCL based approach for handling quality issues in SOLAP systems* (pp. 99–104). Proceedings of the 14th International Conference on Enterprise Information Systems (ICEIS 2012), Wroclaw, Poland.
- Boulil, K., Bimonte, S., & Pinet, F. (2012b). *Un cadre conceptuel basé sur UML et Spatial OCL pour la définition des contraintes d'intégrité dans les systèmes SOLAP* (pp. 42–56). Actes des 8èmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA), Bordeaux, France.
- Cabot, J., Mazón, J.-N., Pardillo, J., & Trujillo, J. (2010). *Specifying aggregation functions in multidimensional models with OCL* (pp. 419–432). Proceedings of the 29th International Conference on Conceptual modeling, Springer-Verlag, Vancouver, BC, Canada.
- Carpani, F., & Ruggia, R. (2001). *An integrity constraints language for a conceptual multidimensional data model*. (pp. 220–227). Proceedings of the Thirteenth International Conference on Software Engineering & Knowledge Engineering, Sheraton Buenos Aires Hotel, Buenos Aires, Argentina, June 13–15, 2001.
- Duboisset, M., Pinet, F., Kang, M. A., & Schneider, M. (2007). A general framework to implement topological relations on composite regions. *Lecture Notes in Computer Science*, 4653, 823–833.
- Eckerson, W. (2000). *Data warehousing special report: Data quality and the bottom line*. TDWI Report Series.
- Ghazzi, F., Ravat, F., Teste, O., Zurfluh, G. (2003, April 22–26). *Constraints and multidimensional databases* (pp. 104–111). Proceedings of the 5th International Conference on Enterprise Information Systems, Angers, France.
- Glorio, O., & Trujillo, J. (2008, September 2–5). *An MDA approach for the development of spatial data warehouses* (pp. 23–32). Proceedings of 10th International Conference Data Warehousing and Knowledge Discovery, Turin, Italy, Springer Berlin/Heidelberg.
- ISO. (2003). ISO 19115:2003 – Geographic information – Metadata. Retrieved from http://www.iso.org/iso/catalogue_detail.htm?csnumber=26020
- JRubik. (2012). Retrieved from <http://rubik.sourceforge.net/jrubik/intro.html>
- Kimball, R. (1996). *The data warehouse toolkit: Practical techniques for building dimensional data warehouses*, Wiley.
- Lenz, H.-J., Shoshani, A. (1997). *Summarizability in OLAP and statistical data bases* (pp. 132–143). Proceedings of the 9th International Conference on Scientific and Statistical Database Management
- Levesque, M., Bédard, Y., Gervais, M., & Devillers, R. (2007, June 14–15). *Towards managing the risks of data misuse for spatial datacubes*. Proceedings of International Symposium on Spatial Data Quality, Enschede, The Netherlands.
- Malinowski, E., & Zimányi, E. (2008). *Advanced data warehouse design, advanced data warehouse design: From conventional to spatial and temporal applications, data-Centric Systems and Applications*. Berlin: Springer.
- Mazón, J.-N., Lechtenböcker, J., & Trujillo, J. (2009). A survey on summarizability issues in multidimensional modeling. *Data & Knowledge Engineering*, 68, 1452–1469.
- Microsoft. (2012). Multidimensional expressions (MDX) reference. Retrieved from <http://msdn.microsoft.com/en-us/library/ms145506.aspx>

- Nazih, H., Boulil, K., Bimonte, S., & Pinet, F. (2012). Demo of the UML-based framework for automatic implemenations of SOLAP systems. Retrieved from <http://eric.univ-lyon2.fr/~sbimonte/SDWTool.mp4>
- No Magic. (2012). MagicDraw. Retrieved from <http://www.nomagic.com/products/magicdraw.html>
- Object Management Group. (2003). MDA Guide, Version 1.0.1. Retrieved from http://www.omg.org/mda/mda_files/MDA_Guide_Version1-0.pdf
- Object Management Group. (2006). Object constraint language OMG, Available Specification, Version 2.0. Retrieved from <http://www.omg.org/spec/OCL/2.0/PDF>
- Object Management Group. (2011). Unified modeling languageTM (OMG UML), Infrastructure, Version 2.4. Retrieved from <http://www.omg.org/spec/UML/2.3/Infrastructure/PDF/>
- Oracle. (2012). Oracle spatial & Oracle locator – Location features for Oracle Database 11 g. Retrieved from <http://www.oracle.com/fr/products/database/options/spatial/index.html>
- Parent, C., Spaccapietra, S., & Zimányi, E. (1999). *Spatio-temporal conceptual models: Data structures + space + time* (pp. 26–33). Proceedings of the 7th ACM international symposium on Advances in geographic information systems, ACM, Kansas City, MO.
- Pedersen, T. B., Jensen, C. S., & Dyreson, C. E. (2001). A foundation for capturing and querying complex multidimensional data. *Information System*, 26, 383–423.
- Pedersen, T. B., Tryfona, N. (2001). *Pre-aggregation in spatial data warehouses* (pp. 460–480). Proceedings of Advances in 7th International Symposium Spatial and Temporal Databases, SSTD.
- Pentaho. (2012). Pentaho analysis services (Mondrian). Retrieved from <http://mondrian.pentaho.com>
- Pinet, F., Duboisset, M., & Schneider, M. (2011). Modelling spatial integrity constraints with OCL. *International Journal of Geomatics and Spatial Analysis*, 21(special issue 2011): 95–123. Lavoisier.
- Pinet, F., Duboisset, M., & Soullignac, V. (2007). Using UML and OCL to maintain the consistency of spatial data in environmental information systems. *Environmental modelling & software*, 22, 1217–1220.
- Pinet, F., & Schneider, M. (2010). Precise design of environmental data warehouses. *Operational Research*, 10, 349–369.
- Ribeiro, L. d. S., Goldschmidt, R. R., Cláudia, M., & Cavalcanti, M. C.. (2011). *Complementing data in the ETL process* (pp. 112–123). Proceedings of the 13th international conference on Data warehousing and knowledge discovery, Springer-Verlag, Toulouse, France.
- Rizzi, S., Abelló, A., Lechtenbörger, J., & Trujillo, J. (2006). Research in data warehouse modeling and design: Dead or alive? (pp. 3–10). *Proceedings of the 9th ACM international workshop on Data warehousing and OLAP*, ACM, Arlington, Virginia, USA.
- Salehi, M. (2009). *Developing a model and a language to identify and specify the integrity constraints in spatial datacubes* (PhD thesis). Faculté des études supérieures de l'Université Laval, Laval..
- Salehi, M., Bédard, Y., Mostafavi, M., & Brodeur, J. (2011). Formal classification of integrity constraints in spatiotemporal database applications. *Journal of Visual Languages & Computing*, 22, 323–339.
- Stefanovic, N., Han, K., & Koperski, K. (2000). Object-based selective materialization for efficient implementation of spatial data cubes. *IEEE Transactions on Knowledge and Data Engineering*, 12, 938–958.