Spatial OLAP and Map Generalization: Model and Algebra

Sandro Bimonte, TSCF, CEMAGREF, France
Michela Bertolotto, University College Dublin, Ireland
Jérôme Gensel, Equipe STEAMER, France
Omar Boussaid, University of Lyon 2, France

ABSTRACT

Map generalization can be used as a central component of Spatial Decision Support Systems to provide a simplified and more readable cartographic visualization of geographic information. Indeed, it supports the user mental process for discovering important and unknown geospatial relations, trends and patterns. Spatial OLAP (SOLAP) integrates spatial data into OLAP and data warehouse systems. SOLAP models and tools are based on the concepts of spatial dimensions and measures that represent the axes and the subjects of the spatio-multidimensional analysis. Although powerful under some respect, current SOLAP models cannot support map generalization capabilities. This paper provides the first effort to integrate Map Generalization and OLAP. Firstly the authors define all modeling and querying requirements to do this integration, and then present a SOLAP model and algebra that support map generalization concepts. The approach extends SOLAP spatial hierarchies introducing multi-association relationships, supports imprecise measures, and it takes into account spatial dimensions constraints generated by multiple map generalization hierarchies.

Keywords: Geographic Information Systems, Map Generalization, Multidimensional Models, Spatial Data Warehouses, Spatial OLAP

1. INTRODUCTION

Map generalization is a process that aims at producing simplified maps at different scales or levels of detail through a set of operators. Map generalization is critical in the spatial decision making process since it allows users to focus on relevant aspects of geographic information ignoring unimportant details. This facilitates the discovery of unknown spatial and thematic relationships and patterns (Vagenont, 2001). Thanks to map generalization operations decision makers can zoom in and-or zoom out into data, filtering geographic information during analysis processes (MacEachren et al., 2004; Cecconi, 2003). Therefore, map generalization can be useful in Spatial Decision Support Systems such as Spatial OLAP.

DOI: 10.4018/jdwm.2012010102

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
Spatial OLAP (SOLAP) tools organize information according to the spatio-multidimensional model (Gomez et al., 2009). SOLAP enables the analysis of numerical and spatial data according to several dimensions, which are organized into thematic and spatial hierarchies. This technology is applied in several application domains (Marketos et al., 2008) (e.g., environmental risk, health, etc.).

Integration of map generalization into SOLAP models could improve the analysis capabilities of spatio-multidimensional operators and also greatly improve the visual component of SOLAP tools (Bédard et al., 2002) by allowing adaptive zoom in/out operations (Ceconi, 2003) on spatial dimensions e.g., adjustment of maps representing spatial dimensions, its contents and the symbolization to target scale in consequence of a zooming operation. Unfortunately, most existing SOLAP models (Ahmed et al., 2006; Damiani et al., 2006; Fidalgo et al., 2004; Glorio et al., 2008; Gomez et al., 2009; Jensen et al., 2004; Malinoswky et al., 2008; Pourras, 2001; Sampaio et al., 2006; Raffaetà et al., 2011), and consequently SOLAP tools (Bimonte et al., 2010; Raffaetà et al., 2011), do not integrate map generalization on complex hierarchies generated by map generalization operators and integrity constraints which define rules to control the multidimensional exploration process.

The contributions of this work are the following.

- It represents the first effort that investigates querying issues in spatial data warehouses integrating map generalization.
- A spatio-multidimensional model, extending (Bimonte et al., 2008), which support previously defined requirements is presented.
- A new and ad-hoc SOLAP algebra is provided that support a constrained exploration of a spatial data warehouse.
- An implementation using a ROLAP architecture is presented that avoids new complex inter spatial data warehouses navigation operators.

In this way, our proposal factors all the requirements for coupling map generalization and SOLAP into a unique logical and implementation framework.

The paper is structured as follows. Section 2 introduces the main concepts of map generalization and Spatial OLAP. Section 3 details the research motivations of our work. Section 4 describes our spatio-multidimensional model and algebra. The implementation is described in Section 5. Related work is presented in Section 6. Discussion and future work conclude the paper.

2. MAP GENERALIZATION AND SPATIAL OLAP

2.1. Map Generalization

Map generalization generates different representations of geographic data at different levels of detail and/or scales (Regnauld et al., 2007) (Figure 1). Map Generalization can support the spatial decision-making process (Vangenot, 2001) since it allows representing spatial phenomena and geographic objects according to different versions. Each of them conveys different information at different levels of abstraction, allowing decision makers to focus only on important decisional aspects and supporting them in the formulation and the verification of hypotheses. Moreover, from a cartographic point of view, map generalization should be used for adaptive zoom operations since it makes maps more readable, facilitating map understanding (Bédard et al., 2002) (Figure 1).

Several map generalization operators have been defined (Regnauld et al., 2007) including: simplification (reduces the number of points representing the geometry), aggregation (merges several geometries), typification (clusters geometries), displacement, etc. They organize geographic objects of maps at different scales or levels of detail in a hierarchical way (Neun et al., 2005; Spaccapietra et al., 2007; Stell et al., 1998) through different kinds of relationships: 1:0 relationships, identity relationships, grouping relationships, and multi-
association relationships. To describe these relationships, we use as an example the generalization of buildings. The relationship $1:0$ eliminates objects (Figure 2b), identity relationships establish a direct link between the same object in the two maps, grouping relationships (Figure 2c) groups several objects in the generalized map, and multi-association relationships establish a link between two groups of objects (Neun et al., 2005; Spaccapietra et al., 2007). Figure 2a shows an example of a multi-association relationship generated by the typification operator: three buildings build1, build2 and build3 are associated with 2 buildings build4 and buildB at the less detailed scale.

In order to integrate map generalization in spatial databases and implement the adaptive zoom, some works define the concept of multiscale database (Devogele et al., 1997) where data sets from different pre-defined scales are stored as features in different tables and linked using foreign keys (e.g., 1:10.000, 1:50.000).

2.2. SOLAP

OLAP and data warehouse (DW) systems facilitate the analysis of huge volumes of data. Data are represented by a multidimensional model (Kimball, 1996). This type of modeling includes dimensions and measures. Dimensions, composed of members, are organized into hierarchies and they represent the analysis axes, while measurements are numerical indicators corresponding to the analysis subjects. Measures
are analyzed at different granularities associated with dimensions hierarchies’ levels using SQL aggregation functions. Common OLAP operators (Kimball, 1996) are: Slice (selection of a part of data warehouse), Dice (elimination of a dimension), Roll-Up (climb to a higher hierarchy level aggregating measure), Drill-Down (inverse of Roll-Up), etc.

Despite the large quantity of spatial data handled by transactional systems and the importance of maps into the spatial decision-making process, traditional OLAP systems do not offer any functionality to handle and exploit spatial data. Consequently, several efforts have been dedicated to the development of Spatial OLAP (SOLAP) (Bédard et al., 2007). SOLAP concepts have been developed to meet this goal. SOLAP models integrate spatial data within dimensions and measures. Spatial dimensions rely on spatial hierarchies defined using spatial or alphanumeric attributes as dimensions (see Gomez et al., 2009 for a survey). Spatial hierarchies are usually complex. For example, it is possible to have non-onto (members skip some levels), non-strict (members have several fathers), and non-covering hierarchies (members have no children) (Malinowsky et al., 2008). These hierarchies define inclusion or intersection relationships between the different spatial levels (Malinowsky et al., 2008; Jensen et al., 2004).

SOLAP systems combine OLAP advanced functionalities for the exploration and analysis of data warehouses, with Geographic Information Systems (GIS) functionalities for storing and visualizing spatial information (Bédard et al., 2007; Raffaetà et al., 2011). Therefore, SOLAP systems synchronize interactive maps, which represent spatial dimensions, with pivot tables and graphic displays to visualize measures and trigger SOLAP operators. An example of SOLAP tool is shown on Figure 3 (Bimonte et al., 2010). It presents GIS and (S)OLAP operators, and the cartographic representations of spatial members and their associated measures.

However, most of existing SOLAP tools support GIS zoom in/out functionalities in classical way by a simple image size reduction,
which can implies not good readable maps for spatial dimensions.

3. INTEGRATION OF MAP GENERALIZATION AND OLAP: SOME ISSUES

Map generalization operators generate maps for different scales or secondary themes, organizing them into a hierarchical complex structure, which are not always characterized by inclusion/intersection topological relationships as classical spatial hierarchies. The introduction of this kind of hierarchies into multidimensional models raises several problems: supporting non-strict and non-covering hierarchies, handling imprecise aggregated measures for spatial members belonging to multi-association relationships, and supporting spatial dimensions constraints.

3.1. Case Study

To explain our approach we present an example of a simulated spatio-multidimensional application concerning domestic accidents (Figure 4a). Dimensions here are: the temporal dimension, the type of accidents (i.e., electricity, gas, etc.) and two spatial dimensions, namely Location and Buildings. Location is described by one spatial hierarchy grouping neighborhoods into cities: neighborhood 1:10,000 < city 1:10,000 means that the spatial level neighborhood 1:10,000 is the child of the spatial level city 1:10,000 in the spatial hierarchy (Figure 4d). Two map generalization hierarchies organize respectively the neighborhoods and cities at two different scales: neighborhood 1:10,000 < neighborhood 1:50,000 (Figure 4c) and city 1:10,000 < city 1:50,000. The spatial dimension Buildings is described by a map generalization hierarchy, which organizes buildings into two scales: building 1:10,000 < building 1:50,000 (Figure 4b). The measure represents the number of accidents and it is aggregated using the sum. In this example, we can analyze domestic accidents per type of accidents, month/year, and buildings of a neighborhood/city at different scales: 1:10,000 and 1:50,000. Here, we use the inclusion topological relationship for the two spatial dimensions since buildings can overlap neighborhoods. The model can answer queries such as: “What is the number of accidents per type of accidents, month and part of building belonging to a neighborhood at the scale 1:10,000?” or “What is the number of accidents per type of accidents, year and part of building belonging to a neighborhood at the scale 1:50,000?”.

3.2. Complex Hierarchies with Multi-Association Relationships and Imprecise Measures

Map generalization hierarchies as described in the previous section can be mapped onto non-covering and non-strict hierarchies. Indeed, the map generalization process can create n:1, n:m (non-strict) and 1:0 relationships. 1:0 relationships can be modeled as non-covering hierarchies where the member with no father skips all the levels of the hierarchy (for example Build6 of the hierarchy of Figure 4b). N:m relationships identify non-strict hierarchies. Moreover, these relationships imply a degree of imprecision associated with the aggregation calculation as map generalization hierarchies 'relax' SOLAP spatial hierarchies (Bimonte et al., 2008). Indeed, unlike traditional SOLAP models (Section 2), hierarchical map generalization relationships do not always involve inclusion or intersection topological relationships between geographic objects at different levels that define classical spatial hierarchies (Malinovsky et al., 2008). This is due to the fact that spatial members are obtained using map generalization operators (such as displacement, etc.). This implies a degree of imprecision associated with the aggregation computation because the contribution of a member to its ancestor might not be quantifiable. In other terms, when a topological relationship exists between father and child spatial members, then it is possible to exactly calculate how much the child intersects the father and use this value to weight measure values during the aggregation process (Jensen et al., 2004). On the other hand,
Figure 4. a) Conceptual model (adapted from Malinowsky et al., 2008) of the SOLAP application, b) Instance of the map generalization hierarchy of buildings, c) Instance of the map generalization hierarchy of neighborhoods, d) Instance of the spatial hierarchy: Neighborhoods < Cities
with map generalization hierarchies this method cannot be applied due to the lack of topological relationship. This means that a degree of imprecision have to be used. Let us explain it.

We assume that in each of the three buildings build1, build2 and build3, 10 gas accidents have occurred during September 1998. Since buildings buildA and buildB are the results of the typification generalization operator (Figure 2a), a multi-association relationship exists between build1, build2 and build3 at the more detailed scale, and buildA and buildB at the less detailed scale.

Thus, when changing the scale through a Roll-Up operation, it is not possible to apply the sum operation to calculate the number of accidents for building buildA. However, it is correct to state that 30 accidents (the sum of the accidents for buildings build1, build2 and build3) have occurred in buildings buildA and buildB during September 1998. Topological relationships between buildings at different scales cannot be used in the aggregation process, because these buildings are not linked by topological inclusion or intersection relationships. As a consequence, it is not possible to safely use classical aggregation functions (i.e., SUM, MIN, MAX, etc.) to model the information represented by measures in generalized maps. Instead, a degree of imprecision is needed (Burdick et al., 2007) for the aggregated measures values.

3.3. Spatial Dimensions Constraints

The introduction of map generalization into multidimensional models requires the definition of one intra spatial dimension constraint on the data model, and of one inter spatial dimensions constraint on SOLAP queries.

3.3.1. Intra Spatial Dimension Constraint

As previously stated, when spatial members of different spatial levels are defined using different scales or levels of detail, then the topological inclusion/overlap relationships could no more be valid. For example, in a spatial data warehouse, if we use only one scale for all the spatial levels of the hierarchy that groups neighborhoods into cities, then neighborhoods are always topologically included into a city (Figure 5a). If we represent cities at a less detail scale (using a map generalization operation), then neighborhoods could not be topologically included into cities anymore (Figure 5b) leading to bad data quality and erroneous SOLAP analysis (Boutil et al., 2010). This problem is called aggregation-generalization mismatch problem (Bédard et al., 2007). It follows that all spatial members of spatial hierarchies must be defined at the same scale or level of detail (Figure 5a). We refer to this constraint as the “Intra spatial dimension constraint”.

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
3.3.2. **Inter Spatial Dimensions Constraint**

As stated previously, using our case study a possible multidimensional query is: "What is the number of accidents per type of accidents, month and building belonging to a neighborhood described at the scale 1:50,000?". This is a well-formed spatio-multidimensional query since spatial levels are represented at the same scale (Figure 6a). If map generalization operators that preserve topology were applied (Bertolotto, 1998), buildings that are included into a neighborhood at the 1:10,000 scale will be also included at the 1:50,000 scale. Therefore, this query is consistent with the topological inclusion relationship associated to spatial dimensions (a building is inside a neighborhood). Let us suppose that a neighborhood at a coarser scale is represented as a point, and that buildings at the most detailed scale are polygons as shown on Figure 6b. In this case, the topological inclusion relationship associated to facts is not verified implying a bad cartographic representation.

This example shows that in the case of multiple spatial dimensions, the scales and/or the levels of detail of the spatial levels used in multidimensional queries have to be the same in order to have a realistic and topologically consistent cartographic representation of spatial members. We refer to this constraint as the "Inter spatial dimensions constraint".

4. **THE DATA MODEL AND ITS ALGEBRA**

In this section, we introduce the spatio-multidimensional model and its algebra.

4.1. **Data Model**

In this section, we define the spatio-multidimensional data model concepts of Geographic
Object, Spatial Hierarchy, Map Generalization Hierarchy and Map Generalization Cube. In order to support reader in understanding our model, we informally map our concepts in UML class diagrams, since defining a UML profile (Glorio et al., 2008) for spatial data warehouses is out of the scope of this work.

4.1.1 Geographic Object

An Object represents an object of the real world. A Geographic Object represents a geographic object.

Definition 1. Object

An Object Structure O is a tuple \( \langle a_1, \ldots, a_n \rangle \) where each \( a_i \) is an attribute defined on a domain \( \text{dom}(a_i) \). An Instance of an Object Structure \( O \) is a tuple \( t = \langle \text{val}(a_1), \ldots, \text{val}(a_n) \rangle \) where \( \forall i \in [1, \ldots, n] \), \( \text{val}(a_i) \in \text{dom}(a_i) \).

Before to define geographic objects, we need to introduce the concept of map generalization attribute as geographic objects can be represented at different scales or levels of detail.

Definition 2. Map Generalization Attribute

A map generalization attribute is an attribute whose domain is a set of elements upon which a total order \( \preceq \) is defined.

Example 1.

A map generalization attribute \( \text{scale} \) can be defined on the domain \( \{1:50.000, 1:10.000\} \) such that \( 1:10.000 \preceq 1:50.000 \), where \( \preceq \) is the total order.

Definition 3. Geographic Object

An Object Structure is a Geographic Object Structure (Figure 7), \( GO \), if:

- The domain of one attribute, \( \text{geom} \) called geometric attribute, is a set of spatial objects (geometries); Let \( g \subset \mathbb{R}^2 \) i.e., a subset of the Euclidian space, then \( \text{dom} (\text{geom}) \in 2^g \).
- A map generalization attribute representing the scale or the level of detail used for its representation.
- All the instances of a Geographic Object Structure have the same map generalization attribute value.

We denote by \( I(GO) \) the set of instances of \( GO \).

Example 2.

The Geographic Object Structure representing the buildings is \( GO_{\text{building}} = \langle \text{geom}, \text{name}, \text{scale} \rangle \) where \( \text{geom} \) is the geometric attribute, \( \text{name} \) is the name of the building, and \( \text{scale} \) is the map generalization attribute (Figure 7). An instance of \( GO_{\text{building}} \) \( ist_1 = \langle p1, \text{build6}, 1:10.000 \rangle \) where \( p1 \) is a geometric object (Figure 2) and \( 1:10.000 \) is the scale used.
4.1.2. Spatial Hierarchy and Map Generalization Hierarchy

Spatial hierarchies and map generalization hierarchies are usually very complex (Section 2). They can be non-covering and non-strict hierarchies. Definition 4 introduces the concept of Spatial Hierarchy which organizes the entities into a hierarchical structure allowing to model complex hierarchies. In particular, the Spatial Hierarchy Structure SH organizes the spatial levels of the hierarchy \( L \cup \{ \top \} \) into a lattice structure through the partial order \( \leq \) (Figure 8). An Instance of Spatial Hierarchy Structure defines a tree of geographic objects through the partial order \( < \) (Figure 4d).

**Definition 4. Spatial Hierarchy**

A Spatial Hierarchy Structure is a tuple \( SH = \langle L, \llbracket \mid \top, \leq \rangle \rangle \) (Figure 8) where:

- \( L \) is a set of Geographic Object Structures \( \{ GO_p,...,GO_d \} \) (Spatial levels)
- \( \llbracket \) is a Geographic Object Structure (Bottom Spatial level)
- \( \top \) is a Geographic Object Structure and it contains one instance called `all` (Top Spatial level)
  - All the Geographic Object Structures belonging to the set \( L \cup \{ \mid \top \} \) have the same map generalization attribute \( \leq \) is a partial order defined on \( L \cup \{ \mid \top \} \) such as:
    - \( \leq \) forms a lattice where \( \mid \) and \( \top \) are respectively the bottom and top of the order

The Geographic Object Structures belonging to \( L \cup \{ \mid \top \} \) are denoted by Levels(SH).

An Instance of a Spatial Hierarchy SH is a partial order \( < \) defined on the instances of the spatial levels of SH, such as:

- if \( t_i < t_j \), then \( GO_i \leq GO_j \), where \( t_i \in I(GO_i) \), \( t_j \in I(GO_j) \) and \( GO_i, GO_j \) are two spatial levels of \( SH \)
- All instances have the same map generalization attribute value (Intra Spatial Dimension Constraint)

The set of leaves of the tree represented by \( < \) with root \( t \) are denoted as Leaves(SH, \( t \)), for example Leaves(MGH\textsubscript{location}, buildC) = \{build4, build5\} (Figure 4b).

The rule defined by a Spatial Hierarchy on the map generalization attribute values with different spatial levels supports the intra spatial dimension constraint (Section 3.3) as it imposes that spatial members of different spatial levels belong to the same scale or level of detail.

**Example 3.**

The Spatial Hierarchy that groups neighbourhods into cities is \( SH_{location} = \langle L_{location}, GO_{buildings}, GO_{all_neigh} \rangle \) where \( L_{location} = \{ GO_{city} (1:10000) \} \) and \( GO_{all_neigh} \) \( ((GO_{city} (1:10000) \leq \langle GO_{city} (1:10000), \langle GO_{city} (1:10000) \leq \langle \langle \leq \rangle \rangle) \) (Figure 8). Its instance is shown in Figure 4d.

**Definition 5. Hierarchy**

A Hierarchy is defined as a spatial hierarchy with the only difference that all levels are Object Structures.

A Map Generalization Hierarchy is a Spatial Hierarchy where all the levels are spatial levels representing the geographic information at different scales or according to secondary themes, and members are the results of the application of map generalization operators to the members of lower levels. Map Generalization Hierarchies extend SOLAP spatial hierarchies by modeling multi-association relationships between members.

**Definition 6. Map Generalization Hierarchy**

A Map Generalization Hierarchy Structure MGH is a Spatial Hierarchy where:

- \( \leq \) forms a total order
- All spatial levels are Geographic Objects whose instances have different map generalization attribute values such that:
  - if \( GO_i \leq GO_j \), then \( I(GO_i).val(scale) \uparrow I(GO_j).val(scale) \) where scale is the map generalization attribute (a coarser
spatial level has a less detailed level or coarser scale.

- A function MultiAssociation: \( I(L(MGH)) \rightarrow 2^{L(MGH)} \) exists which permits to identify spatial members belonging to the same group of a multi-association relationship (Represents multi-association relationships between spatial members).

The total order \( \uparrow \) defined by the map generalization attribute values is called ‘map generalization path’.

### Example 4.

The Map Generalization Hierarchy describing buildings at different scales is \( MGH_{buildingsScale} = \langle L_{buildingsScale}, GO_{building1:10000}, \rangle \) where \( L_{buildingsScale} = \{ \text{GO}_{building1:50000}, \text{GO}_{building1:10000}, \text{GO}_{building:10000}, \text{GO}_{building:50000}, \text{GO}_{building:10000}, \rangle \). Its instance is shown in Figure 4b.

The value of the scale attribute of \( \text{GO}_{building1:10000} \) is 10,000 as it represents the scale 1:10,000 and for \( \text{GO}_{building1:50000} \) is 50,000 as it represents buildings at the scale 1:50,000, then the map generalization path is 10,000 \( \uparrow \) 50,000.

The buildings buildA and buildB are in a multi-association relationship and belong to the same group. Then, MultiAssociation (buildA) = {buildA, buildB} (Figure 2a), and MultiAssociation (buildC) = {buildC} because buildC is not involved in any multi-association relationship (Figure 2c).

It is important to note that the scope of Map Generalization Hierarchies includes only modeling links between layers at different scales or levels of detail. We assume that generalized maps and links are created by cartographic experts or using systems for automatic generalization (Regnauld et al., 2007).

### 4.1.3. Map Generalization Cube

A Map Generalization Cube represents Structure the spatio-multidimensional application where a Map Generalization hierarchy is defined for each spatial level of the spatial hierarchies (Figure 9). A Map Generalization Cube Instance represents the facts table data where all the dimensions are at the most detailed levels.

### Definition 7. Map Generalization Cube

A Map Generalization Cube Structure is a tuple \( MGC=(SH_1, SH_2, H_1, H_2, m_1, m_2) \) (Figure 9) where:

- \( \forall i \in [1, \ldots, p] \), \( H_i \) is a Hierarchy Structure (Dimensions)
- \( \forall j \in [1, \ldots, q] \), \( SH_j \) is a Spatial Hierarchy and (Spatial dimensions)
  - \( \forall \) Spatial Level GO, a Map Generalization Hierarchy MGH exists whose bottom spatial level is GO (Each spatial level of a spatial hierarchy is associated with a map generalization hierarchy).
  - All map generalization hierarchies instances have the same map generalization path
- \( \forall v \in [1, \ldots, n] \), \( m_v \) is a numerical attribute (Measures)

An Instance of a Map Generalization Cube Structure MGC, noted \( I(MGC) \), is a set of tuples \( \{s_1, \ldots, s_p, t_1, \ldots, t_q, \text{val}_1, \ldots, \text{val}_n\} \) where:

- \( \forall i \in [1, \ldots, p] \), \( t_i \) is an instance of the bottom level of \( H_i \) (Members)
- \( \forall j \in [1, \ldots, q] \), \( s_t \) is an instance of the bottom spatial level of \( SH_j \) (Spatial members)
- \( \forall v \in [1, \ldots, n] \), \( \text{val}_v \in \text{dom}(m_v) \) (Measures values)

The hierarchies \( H_1, \ldots, H_p \) of a Map Generalization Cube Structure are non-spatial hierarchies, while \( SH_1, \ldots, SH_q \) are spatial hierarchies. Each level of a Spatial Hierarchy is represented at different scales or levels of detail thanks to a Map Generalization hierarchy. Numerical measures \( m_1, \ldots, m_n \) are defined using precise domains since spatial members of the most detailed level of the spatial hierarchy can not belong to multi-association relationships. An
Figure 9. Study case application: Map generalization cube structure
instance of a Map Generalization Cube Structure represents the facts table data.

**Example 5.**

The Map Generalization Cube Structure of our multidimensional application (Figure 9) is $MGC_{\text{accidents}} = \langle SH_{\text{location}}, SH_{\text{buildings}}, H_{\text{time}}, nb\text{accidents} \rangle$ where:

- $\text{dom}(nb\text{accidents}) = N$, (Measure)
- $SH_{\text{buildings}}$ (Spatial Hierarchy)
  - $GO_{\text{building1:10000}}$ of $SH_{\text{buildings}}$ is associated with $MGH_{\text{buildingsScale}}$
- $SH_{\text{location}}$ (Spatial Hierarchy)
  - $GO_{\text{city1:10000}}$ of $SH_{\text{location}}$ is associated with $MGH_{\text{cityScale}}$
  - $GO_{\text{neighb1:10000}}$ of $SH_{\text{location}}$ is associated with $MGH_{\text{neighbScale}}$ (due to space limitation, we do not provide formal details for all the hierarchies).

Here, $MGC_{\text{accidents}}$ defines four hierarchies $SH_{\text{building}}, H_{\text{accidents}}, H_{\text{time}}, SH_{\text{location}}$ and one measure $nb\text{accidents}$. Moreover, it associates a Map Generalization hierarchy with each level of the spatial hierarchies. These Map Generalization hierarchies have the same generalization path. Table 1 shows the instance of $MGC_{\text{accidents}}$. It allows answering queries using the most detailed level of each hierarchy such as “How many accidents occurred per month, type of accident and part of building belonging to a neighborhood at the 1:10.000 scale?” No aggregation function is needed.

### 4.2. Multidimensional Queries

In this section, we present the concept of Map Generalization Query. A Map Generalization Query represents a multidimensional query using non-spatial levels and a spatial level for each Map Generalization Hierarchy. Map Generalization Queries introduce an imprecise representation of aggregated measures (imprecise domains). The model, using user-defined functions, calculates imprecise measures values for spatial members belonging to multiassociation relationships. Moreover, a Map Generalization Query satisfies the inter spatial dimensions constraint by defining a constraint on spatial levels.

In order to simplify the definition of the Map Generalization Query we introduce the notation of ComposedLeaves.

**Definition 8.**

Let $MGC$ be a Map Generalization Cube, and $MGH$ be a Map Generalization Hierarchy associated with a spatial level $GO_{\text{spatial}}$ of a Spatial hierarchy $SH$ ($MGH = (SH, GO_{\text{spatial}})$).

<table>
<thead>
<tr>
<th>Neighb1:10.000</th>
<th>Building1:10.000</th>
<th>Month</th>
<th>Type</th>
<th>nbaccidents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neighb1</td>
<td>Build1</td>
<td>9-1998</td>
<td>gas</td>
<td>1</td>
</tr>
<tr>
<td>Neighb1</td>
<td>Build1</td>
<td>10-1998</td>
<td>gas</td>
<td>1</td>
</tr>
<tr>
<td>Neighb1</td>
<td>Build2</td>
<td>10-1998</td>
<td>gas</td>
<td>1</td>
</tr>
<tr>
<td>Neighb1</td>
<td>Build3</td>
<td>10-1998</td>
<td>gas</td>
<td>1</td>
</tr>
<tr>
<td>Neighb1</td>
<td>Build2</td>
<td>3-1999</td>
<td>fire</td>
<td>6</td>
</tr>
<tr>
<td>Neighb1</td>
<td>Build4</td>
<td>3-1999</td>
<td>fire</td>
<td>2</td>
</tr>
<tr>
<td>Neighb1</td>
<td>Build5</td>
<td>3-1999</td>
<td>fire</td>
<td>0</td>
</tr>
<tr>
<td>Neighb1</td>
<td>Build6</td>
<td>10-1998</td>
<td>gas</td>
<td>3</td>
</tr>
</tbody>
</table>

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
The set of leaves of the tree defined by the union of the instance of SH and the instance of MGH and having as root an instance st of a spatial level MGH is ComposedLeaves(MGH, SH, st).

ComposedLeaves(MGH, SH, st) is equal to \{(Leaves(SH, st1) \cup ... \cup Leaves(SH, stl)) where st1, ... sl belong to Leaves (MGH, st).

This definition allows to merge a spatial hierarchy instance and a map generalization hierarchy instance in the same tree. This allows finding ancestors of a spatial member belonging to a map generalization hierarchy in the tree.

Example 6.

ComposedLeaves(MGH_{CityScale}, SH_{location}, CityA) = {Neighb1, Neighb2}, since Leaves(MGH_{CityScale}, CityA) = City1 and leaves(SH_{Location}, City1) = {Neighb1, Neighb2}.

We can now formalize the Map Generalization Query.

Definition 9. Imprecise domain

An imprecise domain impreciseDom over a domain dom is a non empty subset of the power set of dom. Elements of impreciseDom are called imprecise values (Burwick et al., 2007).

Definition 10. Map Generalization Query

- A Map Generalization Query Structure \(Q\) is a tuple \(\langle MGC, L, \Theta, \pi \rangle\) (Figure 11a) where:
  - \(MGC = \langle S_{H_1}, ..., S_{H_p}, H_{p}, ..., H_{q}, m_{p}, ..., m_q \rangle\) is a Map Generalization Cube Structure (Spatial data warehouse)
  - \(L\) is a tuple of Object Structures and Geographic Object Structures \(\langle O_{r1}, ..., O_{rk} \rangle\) such that:
    - \(\forall j \in [1, ... q] GO_j\) is a spatial level of the Map Generalization Hierarchy \(MGH_j\), associated with a spatial level of \(SH_j\) (Spatial levels of map generalization hierarchies used in the query)
    - \(GO_{p}, ..., GO_{q}\) have the same map generalization attribute value (Inner spatial dimensions constraint)
    - \(\forall i \in [1, ... p] O_i\) is a level of \(H_i\) (Levels of classical hierarchies used in the query)
    - \(\pi\) is an optional predicate defined on some (spatial) levels of \(L\)
    - \(\Theta = \langle f_{r1}, ..., f_{r n} \rangle\) where \(\forall v \in [1, ... n], f_r : \text{dom}(m_r) \rightarrow \text{dom}(m_r)\)
    - \(\Theta\) are the aggregation functions applied to the measures \(m_1, ..., m_q\)
    - \(\gamma = \langle \text{imprecise}_{r1}, ..., \text{imprecise}_{rn}\rangle\) where \(\forall v \in [1, ... n], \text{imprecise}_v : \text{dom}(m_r) \times \text{dom}(GO_r) \times ... \times \text{dom}(GO_n) \rightarrow \text{impreciseDom}(m_r)\) (Functions to calculate imprecise aggregated measures associated with spatial members belonging to multi-association relationships)

An instance of a Map Generalization Query Structure \(Q\), denoted as \(L(Q)\), is the set of all possible tuples \(\langle st_{r1}, ..., st_{r n}, t_{p}, ..., t_{q}, val_{r1}, ..., val_{rn} \rangle\) (Figure 10) where:

- \(\forall i \in [1, ... p] t_i\) is an instance of \(O_i\) that satisfy the condition \(\pi\) (Members)
- \(\forall j \in [1, ... q] st_j\) is an instance of \(GO_j\) that satisfy the condition \(\pi\) (Spatial members)
- \(\forall j \in [1, ... q] \text{if MultiAssociation}(st_j) = \{st_j\}\)
  - then \(\forall v \in [1, ... n] val_{ry} = f_r(\text{val}_{r1}, ..., \text{val}_{rn})\) (aggregation of measure values associated with members that do not belong to multi-association relationships)
  - else \(\forall v \in [1, ... n] val_{ry} = \text{imprecise}_v(f_r(\text{val}_{r1}, ..., \text{val}_{rn}), st_{p}, ..., st_{q})\) (imprecise aggregated measure values associated with members that belong to multi-association relationships)

where:

- \(\forall z \in [1, ... k] \text{val}_{rz}\) belongs to a tuple \(\langle \text{val}_{r1}, ..., \text{val}_{rn}, \text{val}_{z}, ..., \text{val}_{zn} \rangle\) of \(L(MGC)\) such that:

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
Figure 10. Map generalization query instance

Example 7.

The Map Generalization Query $Q_{\text{accidents}_1:50000}$ = \langle MGC_{\text{accidents}}, \langle GO_{\text{neigh1:50000}}, GO_{\text{building1:50000}} \rangle, O_{\text{year}}, O_{\text{all accidents}}, \langle \text{SUM} \rangle, \langle \text{imprecise}_{\text{nbAccidents}} \rangle \rangle$ represents the number of accidents per year, type of accident and part of building belonging to a neighborhood at the 1:50,000 scale (Figure 11a).

$Q_{\text{accidents}_1:50000}$ is composed of:

- $MGC_{\text{accidents}}$ which represents the fact table,
- $GO_{\text{neigh1:50000}}, GO_{\text{building1:50000}}$ the levels used in the multidimensional query,
- The aggregation function $\text{SUM}$ used to aggregate the measure $\text{nbAccidents}$ defined by the Map Generalization Cube,
- A function $\text{imprecise}_{\text{nbAccidents}}$ defined on $\text{dom}(\text{nbAccidents}) \times \text{h}(\text{GO}_{\text{building1:50000}})$ that returns a range value for the members belonging to a multi-association relationships.

The instance of $Q_{\text{accidents}_1:50000}$ is shown in Table 2. The measure $\text{nbAccidents}$ presents precise and range values. For instance, the value of $\text{nbAccidents}$ for the building $\text{buildC}$ on 1999 and for all types of accidents is the sum of the measure values of the Map Generalization Cube associated to: $\text{build4}, \text{build}, \text{fire}, \text{gaz}$ and $\text{3-1999}$, since $\text{buildC}$ (Figure 2c) is the aggregation of $\text{build4}$ and $\text{build5}$ (group relationship) ($\text{ComposedLeaves}(\text{MG} H_{\text{buildingScale}}, \text{SH}_{\text{building}}, \text{buildC}) = \{\text{build4, build5}\}$, 3-1999) is the child of 1999 ($\text{Leaves}(\text{H}_{\text{month}}, 1999) = \{3-1999\}$), and fire and gaz belong to all types of accidents ($\text{Leaves}(\text{H}_{\text{accidents}_1:50000}, \text{all}) = \{\text{gaz, fire}\}$). On the contrary, as $\text{buildA}$ and $\text{buildB}$ belong to the same group in a multi-association relationship, then on 1998 their imprecise value is the range value [0-4] because they are associated with the buildings $\text{build1}$, $\text{build2}$ and $\text{build3}$ at the less detailed scale (Figure 2a and Table 1).

The query satisfies the inter spatial dimensions constraint because the spatial levels $\text{GO}_{\text{neigh1:50000}}$ and $\text{GO}_{\text{building1:50000}}$ refer to the same scale.

It is important to note that our model does not impose any constraint on the functions for calculating imprecise measures. For example, users can define a particular estimation function based on their expertise to calculate imprecise measures and therefore restrict the range values.

Finally, the model avoids using aggregated results to answer multidimensional queries since these could be imprecise values. Indeed, the model always uses facts table data (detailed measures), represented by the Map Generalization Cube Instance, to calculate Map Generalization Query instance.
4.3. Spatio-Multidimensional Operators

In this section, we present the operators for exploring the spatial data warehouse. We define the Slice operator, which selects a sub-set of spatial/alphanumeric members, and two drill operators allowing navigating into spatial and classical dimensions. These operators are applied to a Map Generalization Query and result in another Map Generalization Query. This means that the algebra is closed.

4.3.1. Slice

The Slice operator cuts a part of the spatial data warehouse. It takes as input a Map Generalization Query (Q) and a predicate (σ) defined on some levels. It returns another Map Gener-

<table>
<thead>
<tr>
<th>Neighbor 1: 50,000</th>
<th>Building 1: 50,000</th>
<th>Year</th>
<th>Type</th>
<th>nbAccidents</th>
</tr>
</thead>
<tbody>
<tr>
<td>NeighA</td>
<td>buildB</td>
<td>1998</td>
<td>All</td>
<td>[0, 4]</td>
</tr>
<tr>
<td>NeighA</td>
<td>buildA</td>
<td>1998</td>
<td>All</td>
<td>[0, 4]</td>
</tr>
<tr>
<td>NeighA</td>
<td>buildA</td>
<td>1999</td>
<td>All</td>
<td>[0, 6]</td>
</tr>
<tr>
<td>NeighA</td>
<td>buildB</td>
<td>1999</td>
<td>All</td>
<td>[0, 6]</td>
</tr>
<tr>
<td>NeighA</td>
<td>buildC</td>
<td>1999</td>
<td>All</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 2. Instance of \( Q_{\text{accidents:50000}} \): Accidents per year and part of building belonging to a neighborhood at the 1:50000 scale
alization Query with the same structure, but containing only the tuple instances satisfying the predicate.

**Definition 11. Slice**

Let $Q = \langle MGC, L, \Theta, [\pi] \rangle$ a Map Generalization Query Structure, and $\pi$ a condition expressed on some levels of $L$ then $\text{Slice}(Q)[\pi] = Q' = \langle MGC, L, \Theta, \pi \rangle$

**Example 8.**

Let us suppose that the user is only interested in accidents for the building $\text{BuildA}$ per year and all type of accidents. It applies the slice operator at $Q_{\text{accidents1.50000}}$ (Example 7). $\text{Slice}(Q_{\text{accidents1.50000}}) [H(\text{GO}_{\text{buildings1.50000}}) = \text{[BuildA]}] = Q'_{\text{accidents1.50000}}$. The instance of $Q'_{\text{accidents1.50000}}$ contains only the first and the third rows of Table 2.

**4.3.2. Drill operators**

The algebra includes two drill operators:

- **Drill** operators that allow navigating the spatial (Spatial Drill) and classical hierarchies (Thematic Drill).
- **One Zoom** operator that allows changing the scale or the level of detail. It permits to navigate all map generalization hierarchies of the spatial dimensions.

In the following we define these operators in detail.

**4.3.2.1. Thematic Drill**

**Definition 12. Thematic Drill**

Let $Q = \langle MGC, L, \Theta, [\pi] \rangle$ a Map Generalization Query Structure, and $O$ an Object Structure of the hierarchy $H$ of MGC, then $\text{DrillT}(Q)[O] = Q' = \langle MGC, L', \Theta \rangle$

where:

- $L'$ is the same as $L$ except for the level $O$, that replaces the level $O_i$ belonging to $H_i$.

**Example 9.**

Let $Q_{\text{accidentsfacts}} = \langle MGC_{\text{accidents}}, \langle \text{GO}_{\text{buildings1.10000}}, \text{GO}_{\text{month1.10000}}, O_{\text{type}}, \langle \text{SUM}, \langle \text{imprecise}_{\text{accidents}} \rangle \rangle \rangle$ represent the fact table data (accidents per month, part of building belonging to neighborhood at the 1:10.000 scale and types of accidents). Its instance is represented in Table 1. The user can navigate through the year level and all types of accidents using the Thematic Drill operator twice. $\text{DrillT}(\text{DrillT}(Q_{\text{accidentsfacts}}) [S_{\text{year}}]) [S_{\text{all accidents}}] = Q_{\text{accidents}}$ (Figure 12). Its instance is shown on Table 3.

**4.3.2.2. Spatial Drill**

**Definition 13. Spatial Drill**

Let $Q = \langle MGC, L, \Theta, [\pi] \rangle$ a Map Generalization Query Structure, and $GO$ a Geographic Object Structure of the spatial hierarchy $SH_i$ of MGC then $\text{DrillS}(Q)[GO] = Q' = \langle MGC, L', \Theta \rangle$ where $L'$ is the same as $L$ except for:

- GO replaces GO,
- The other spatial levels are replaced by the bottom spatial level of their map generalization hierarchies;

This operator allows navigating into a spatial hierarchy and in order to grant the inter spatial dimension constraint it sets all the spatial hierarchies at their bottom level.

**4.3.2.3. Zoom**

The zoom operator takes a Map Generalization Query ($Q$) and a value of a map generalization attribute value called MapGenLevel as input and returns another Map Generalization Query with the same levels of $Q$ except for the spatial levels of the spatial dimensions. The spatial levels of $Q$ are replaced by the spatial levels with the map generalization values equal to MapGenLevel.

The constraint on map generalization paths of the Map Generalization Cube grants that a spatial level with MapGenLevel exists for each map generalization hierarchy.
Definition 14. Zoom

Let $Q = \langle \text{MGC}, L, \Theta, [\pi] \rangle$ a Map Generalization Query Structure, and MapGenLevel be the map generalization attribute value of a Geographic Object Structure belonging to one Map Generalization Hierarchy of MGC then \( \text{Zoom}(Q) \) \( [\text{MapGenLevel}]= Q' = \langle \text{MGC}, L', \Theta \rangle \) where \( L' \) is the same as \( L \) except for the spatial levels \( G_{O'} \), ..., \( G_{O'} \), ..., \( G_{O_q} \) such that their map generalization attribute value is MapGenLevel and \( \forall j \in [1, \ldots, q] \) \( G_{O'} \) belongs to the same Map Generalization Hierarchy of \( G_{O_j} \)

Example 10.

To analyze the accidents that have occurred in each building at 1:50,000 scale per year and all types of accidents, the user applies the Zoom operator at the Map Generalization Query of the previous example (Figure 12) as follows: \( \text{Zoom}(Q_{\text{accidents}}[1:50,000]) = Q_{\text{accidents}[1:50,000]} \) (Figure 11). This operator changes the scale of the two spatial dimensions climbing to scale 1:50,000 in the Map Generalization Hierarchies associated with buildings and neighborhoods.

Table 3. Instance of $Q_{\text{accidents}}$: Accidents per year and part of building belonging to a neighborhood at the 1:10000 scale

<table>
<thead>
<tr>
<th>Neigh1:10.000</th>
<th>Building1:10.000</th>
<th>Year</th>
<th>All_type</th>
<th>nbaccidents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neigh1</td>
<td>Build1</td>
<td>1998</td>
<td>All</td>
<td>2</td>
</tr>
<tr>
<td>Neigh1</td>
<td>Build2</td>
<td>1998</td>
<td>All</td>
<td>1</td>
</tr>
<tr>
<td>Neigh1</td>
<td>Build3</td>
<td>1998</td>
<td>All</td>
<td>1</td>
</tr>
<tr>
<td>Neigh1</td>
<td>Build3</td>
<td>1999</td>
<td>All</td>
<td>6</td>
</tr>
<tr>
<td>Neigh1</td>
<td>Build4</td>
<td>1999</td>
<td>All</td>
<td>5</td>
</tr>
<tr>
<td>Neigh1</td>
<td>Build5</td>
<td>1999</td>
<td>All</td>
<td>0</td>
</tr>
<tr>
<td>Neigh1</td>
<td>Build6</td>
<td>1998</td>
<td>All</td>
<td>3</td>
</tr>
</tbody>
</table>
5. IMPLEMENTATION

In this section, we present the implementation of our model and algebra on a data sample in order to show the feasibility of our approach. We use a ROLAP architecture with Oracle Spatial as spatial data warehouse tier, Mondrian as OLAP server tier, and JPivot as OLAP client (Pentaho, http://www.pentaho.com).

First, we introduce the relational data model of the spatial data warehouse and the corresponding OLAP Server model. Then, we describe the implementation of inter and intra spatial dimensions constraints in JPivot and Oracle respectively. Finally, we introduce the implementation of the algebra operators.

5.1. Multidimensional Data Model

The relational data model used in our implementation is depicted in Figure 13. We use a star schema to model our spatial data warehouse (Kimball, 1996). The facts table is “Accidents”. It contains four foreign keys to the dimensions. The facts table has one measure “nbaccidents” that is the number of accidents per building, neighborhood, month and type of accident. Note that, according to our definition of Map Generalization Cube (Section 4.1), it contains only precise measures. Imprecise measures are taken into account only in the OLAP Server tier (as detailed in the next section).

Spatial dimensions contain several hierarchies. We choose to represent each hierarchy with a separate table. Let us describe the spatial and map generalization hierarchies. The spatial hierarchy “H_{location}”, which groups neighborhood into cities, is represented by the table “Neighborhoods”. The spatial hierarchy “H_{buildings}” is represented using only the first level of the table “Buildings”. According to Glorio et al. (2008), each spatial level of the spatial hierarchies has the geometric attribute “geom” (that in Oracle...
Spatial (http://www.oracle.com, 2011) is represented by the SDO_GEMETRY type, and an attribute that represents the scale “Scale” according to the definition of Geographic Object (Section 4.1).

Map generalization hierarchies are more complex since they could have two supplementary attributes: “multiassociation” and “weight” (Figure 14). “Multiassociation” expresses whether the spatial member belongs to a multi-association relationship and its belonging group. It corresponds to the Multiassociation function of our data model (cf. Section 4.2). For example, for the buildings BuildA and BuildB this property is “1” because they belong to a group identified with “1”. “Weight” is used to avoid multiple counts for spatial members as most detailed spatial members are duplicated since they belong to multi-association relationships (see Bimonte et al., 2008 for more details).

Our model comprises also three Map Generalization hierarchies: “H_buildingscale” for buildings that are represented by the table “Buildings”, “H_neighborhoods” for neighborhoods that are represented by the table “Neighbs”, and “H_citiescale” for cities represented using the table “Cities”. Only “Buildings” has the “Multiassociation” and “Weight” attributes since it is the only hierarchy whose instance presents multi-association relationships (Figure 14).

Our approach has been implemented using the ROLAP Server Mondrian (Pentaho, http://www.pentaho.com). Figure 15 shows the Mondrian multidimensional model of the case study example. We define two spatial dimensions LocationDimension and BuildingsDimension. BuildingsDimension is described by a map generalization hierarchy BuildingsScale and a spatial hierarchy Buildings composed of one spatial level Building10000 in order to conform with the formal model which defines a spatial hierarchy for each spatial dimension. The level property MultiAssociation is used for the levels of the BuildingScale hierarchy since its instance presents multi-association relationships. The map generalization attribute of all spatial levels is represented using the level property Scale. The spatial dimension LocationDimension is described by a spatial hierarchy (Location) and two map generalization hierarchies (Neighborhoods and CitiesScale).

The bottom range value of the imprecise measure is defined by means of a derived measure (LowNbAccidents) defined as:

\[ IIF({(\{BuildingsDimension.BuildingsScale\}.CurrentMember.
Properties("MultiAssociation") = "0"},
\{Measures\}.\{TopNbAccidents\}, 0.0). \]

This derived measure is valued to 0 if the value of MultiAssociation is not 0 (the current spatial member belongs to a multi-association relationship); else its value is TopNbAccidents, which is the exact measure. In this work, we have used a simple “if” statement and a member property to represent the function (impreciseidents) in order to calculate imprecise measures (LowNbAccidents). If the function is more complex, then the range values should be

Figure 14. Buildings table

![Figure 14. Buildings table](image_url)
defined using two derived measures calculated by means of user-defined Java functions.

5.2. Inter and Intra Spatial Dimensions Constraints

Intra spatial dimension constraints are similar to inter levels integrity constraint defined in Boulii et al. (2010). Then, we use the same approach of Boulii et al. (2010) that consists in using SQL triggers that avoid inserting different map generalization attribute values.

Let us consider the spatial hierarchy Location (table “Neighscities of Figure 13), for first user defines using the SQL statement CHECK a constraint on one scale attribute (CHECK (SCALEN = SCALEN), then using a generic
trigger she/he controls that all the scale attributes have the same values (Figure 16).

The inter spatial dimensions constraint is verified at each Spatial OLAP query: all members of the spatial dimensions have to be associated with the same scale attribute. We have defined a Java method in the JSP page of the OLAP client JPiviot (using the com.tonbeller.jpivot.olap. query.DrillReplaceExt) method to represent an OLAP query in Mondrian) that parses each result (Pivot table) of the OLAP queries and verifies that all the spatial elements of the queries have the same scale. If a query does not respect this constraint the last “good” query is loaded. For example, user selects using the Cube Navigator of JPiviot the neighborhood Neigh1 and the building BuildA. Using our algorithm, an error message is shown in the browser and the previous query is re-loaded since spatial members do not respect the inter spatial dimension constraint (scale 1:10.000 for Neigh1 and scale 1:50.000 from BuildA).

5.3. Algebra

In order to implement the algebra, we have extended functionalities of the JPiviot OLAP client (Pentaho, http://www.pentaho.com). Drill and Slice operators are natively implemented by JPiviot-Mondrian. On the other hand, in order to correctly implement the Zoom operator we have defined a Java method that replaces each spatial member with its children (using the method com.tonbeller.jpivot.olap.query.DrillReplaceExt) drillExt), drillDown(currentMember) n times on the previous obtained query), if it is a Zoom in operation, or with the spatial members of the higher level if it is a Zoom out operation (using the method (com.tonbeller.jpivot.olap.query.DrillReplaceExt)drillExt).drillUp(hier)).

Let us consider the query that represents accidents per neighborhood and building at the 1:10.000 scale as shown on Figure 17a. Then, the user applies the Zoom out operator by clicking on the button in the user interface to move to the scale 1:50.000. The result is shown on Figure 17b. It represents accidents per buildings and neighborhoods at the 1:50.000 scale. Note that we have used a simulated cartographic representation in Figure 17.

6. RELATED WORK

Some spatio-multidimensional models have been proposed so far to take into account different aspects of spatial information such as multi-granularity, continuous field data, etc. However, no work supports all the previously described requirements for integrating map generalization in SOLAP as described in the following.

Map generalization represents a particular point of multi-representation (Bédard et al., 2002). Thus, the definition of spatial members at different scales or levels of details can be considered as a multirepresentation of the spatio-multidimensional data.

Therefore, from this point of view the most similar work to our proposal are Bédard et al. (2002), Bérnier et al. (2007), Gascueña et al. (2009), and Damiani et al. (2006). The work of Bérnier et al. (2007) uses the multidimensional paradigm to provide on-Demand multi-scale maps. This approach models maps features at different scales by using spatial hierarchies,
but it does not contain measures. The approach of Bédard et al. (2002) defines a UML-based conceptual model that integrates multiple geometric and semantic representations properties of spatial levels. On the other hand, this work does not present a complete multidimensional model with facts and hierarchies.

Moreover, Bédard et al. (2002) suggests (without providing details) modeling multi-representation using different spatial data warehouses and use them to produce other ones. Therefore, changing the representation corresponds to move to another spatial data warehouse. Although the constellation schema described in Abello et al. (2002) and Moody et al. (2000) seems suitable, no further work formally and practically investigates using this approach by specializing data warehouse navigation operators and multidimensional data structures for multi-representation or adaptive
zoom on several spatial dimensions. However, contrary to the idea to navigate among data warehouses, in our approach we integrate multi-representation (map generalization hierarchies) in the form of OLAP hierarchies. Indeed, we exploit OLAP system functionalities to calculate measures values for the different levels of details or scales, and navigate into the spatial data warehouse. In other words, changing the scale or level of detail (adaptive zoom operation) corresponds to a Roll-up or Drill-Down operation on each spatial dimension. In this way, we do not need implement new complex inter data warehouses navigation operators by adding complex programming functionalities to the OLAP system. We allow users to use the map generalization hierarchy structures and instances as classical hierarchies taking benefit of advanced OLAP server functionalities (MDX support, materialized views, etc.).

Gascuena et al. (2009) propose a conceptual model with a multi-representation of spatial members, but they do not support map generalization on several points: imprecise measures and multi-association relationships. They propose a physical schema (relational tables) but any operator to change representation or an implementation into a classical ROLAP architecture are presented.

In the same line, McGuire et al. (2008) define a snowflake schema for an environmental application where three dimensions represent the same spatial members at different resolutions. However, they partially implement the adaptive zoom operator, as they do not support all modeling requirements of map generalization hierarchies (imprecise and multiple spatial dimensions).

Finally, in the same direction Damiani et al. (2006) propose a model where the spatial measure can be represented at different scales or granularities. However, the operator that changes the granularity of the measure does not handle the inter spatial dimensions constraint, and no implementation is provided.

Let us now consider related work on complex hierarchies, integrity constraints and imprecise measures.

Non-strict and non-covering hierarchies are supported by almost all the models as it is a mandatory requirement for multidimensional models for complex data (Pedersen et al., 2001). However, as a consequence of topological inclusion/intersection constraints of spatial hierarchies of existing SOLAP models, imprecise aggregated measures are generally not supported, with the exception of the work by Jensen et al. (2004). They build their model upon work of Pedersen et al. (2001) that handles imprecise measures. However, no model differentiates multi-association relationships from non-strict hierarchies. Indeed, for multi-association relationships an imprecision hierarchical relationship is needed as it is not always possible to use a distribution factor to split measures values to father members (Jensen et al., 2004; Malinowsky et al., 2008) (e.g., even if the sum is 4, the imprecision implies that the aggregate measure value is [0;4]). At the same time, it must be acknowledged that spatial members belonging to the multi-association relationship share the same aggregated measure value (e.g., this is like an implicit distribution factor stating that a spatial member can have a value between 0 and 4 but if we aggregate using the sum, the sum of all the spatial members has to be 4). This is achieved in our model by applying the Multiasociation function.

In Salehi et al. (2008), integrity constraints for spatial data warehouses are investigated. They identify a set of inter- and intra-dimension constraints. They provide a formal spatio-multidimensional model and a hybrid natural language to express constraints. However, no multidimensional operators and implementation have been proposed. On the other hand, from a conceptually point of view, among those constraints, they define the aggregation-generalization problem (internal spatial dimension constraint) and the hypercellularity constraints that could be considered a general case of our inter spatial dimension constraint. Almost all existing SOLAP models implicitly support this constraint since they use a unique scale. They define the topological inclusion/intersection (Malinowsky et al., 2008) or partition (Pour-
rabas, 2003) constraints among spatial members of different levels. Among SOLAP models, only (Malinowsky et al., 2008) provides a conceptual model that defines topological constraints between most detailed levels members of the spatial dimensions. However, this is not sufficient for map generalization as the inter spatial dimension constraint is additional to topological constraints and it is valuable also for coarser levels. Indeed, the inter spatial dimensions constraint is equivalent to the choice of a particular representation in the model proposed by Bédard et al. (2002). In the context of OLAP, some multidimensional models support dimensions constraints. They define constraints on dimension members to support summarizability (Hurtado et al., 2005), constraints among members of different dimensions, and constraints among members of different hierarchies of the same dimension (Ghozzi et al., 2003). However, constraint defined in Ghozzi et al. (2003) is not sufficient for map generalization hierarchies as it forces the members of the two dimensions to not be associated with the same fact.

To conclude, we have discussed how existing SOLAP and OLAP models address only some of issues for map generalization integration, but no model fully supports all these requirements. Our work opens up some interesting and new ways of research for the effective integration of map generalization in SOLAP models and tools. However, our proposal presents several limitations. Firstly, we do not investigate summarizability problems related to non-strict hierarchies as we use always facts table data to calculate aggregated measures. In this way, we also avoid to use imprecise values to calculate measures. On the contrary, interval arithmetic (Burdick et al., 2007) would be necessary for SOLAP queries using materialized views.

7. CONCLUSIONS AND FUTURE WORK

Map generalization allows representing geographic data at different scales and/or levels of detail, which is useful for spatial decision-making process. On the other hand, Spatial OLAP models and systems do not currently support map generalization. This is an important limitation since map generalization hierarchies could enhance analysis capabilities of SOLAP models. The integration of map generalization and multi-representation within SOLAP raises several problems since map generalization hierarchies are complex and they are characterized by multi-association relationships. Moreover, spatial dimensions constraints have to be defined on the multidimensional data models and on Spatial OLAP queries.

In this paper, we extend the spatio-multidimensional data model (Bimonte et al., 2008) with complex modeling properties, and an associated new and ad-hoc algebra to correctly support the issues described above. Our model extends the concept of spatial hierarchies defining map generalization hierarchies that explicitly represent multi-association relationships between spatial members. The multidimensional model provides an imprecise representation of aggregated measures values for spatial members belonging to these relationships by means of user-defined functions. The multidimensional model supports spatial dimensions constraints on data and queries. A formalization of the associated algebra is also introduced in this paper that includes classical slice and drill operators. The algebra provides also a new operator (Zoom) to climb all map generalization hierarchies on the same scale or level of detail respecting inter spatial dimensions constraint. The Zoom operator allows supporting adaptive zoom in SOLAP tools.

The model and the algebra have been implemented using an extension of the ROLAP Server Mondrian, and of the OLAP Client Jpivot.

Our work is now directed towards the support for the cartographic visualization of our model using the prototype GeWOlap (Bimonte et al., 2010) as visualization is mandatory for an effective SOLAP analysis (Raffaetà et al., 2011).

Performance studies are also needed. Performance depends on the complexity of the model for the computation of imprecise...
measures. In detail, in our case study as the imprecise function simply sets ‘0’ for the lower bound of the imprecise measure it does not affect performance, but the case when imprecise functions use spatial operators (such as the area, the perimeter, etc.) has to be studied. Finally, using materialized views with numeric intervals should be studied.

ACKNOWLEDGMENT

Research presented in this paper was funded by a Strategic Research Cluster grant (07/SRC/11168) by Science Foundation Ireland under the National Development Plan. The authors gratefully acknowledge this support.

REFERENCES


Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.


Sandro Bimonte was born in 1978, he is researcher at CEMAGREF, and more exactly he is at TSCF. He obtained his PhD at INSA-Lyon, France (2004-2007). From 2007-2008, he carried out researches at IMAG, France. He is Managing Editor of Journal of Decision Systems, Editor Board member of International Journal of Decision Support System Technology, and International Journal of Data Mining, Modelling and Management and member of the Commission on GeoVisualization of the International Cartographic Association. His research activities concern spatial data warehouses and spatial olap, visual languages, geographic information systems, spatio-temporal databases and geovisualization.

Michela Bertolotto received a PhD in computer science from the University of Genoa (Italy) in 1998. Subsequently she worked as a postdoctoral research associate in the National Center for Geographic Information and Analysis (NCGIA) at the University of Maine. Since 2000 she has been a faculty member at the School of Computer Science and Informatics of University College Dublin. Her research interests include web-based and wireless GIS, spatio-temporal data modelling and 3D interfaces.

Jérôme Gensel is a full professor at the University Pierre Mendès France of Grenoble, France, since 2007. He received his PhD in 1995 from the University of Grenoble for his work on constraint programming and knowledge representation in the Sherpa project at the French National Institute of Computer Sciences and Automatics (INRIA). He joined the Laboratory of Informatics in Grenoble (LIG, formerly called LSR-IMAG Laboratory) in 2001. His research interests include representation and inference of spatio-temporal information, ontologies and knowledge representation, geospatial semantic web, and ubiquitous geographical information systems.

Omar Boussaid is a full professor in computer science at the school of economics and management of the University of Lyon 2, France. He received his PhD in computer science from the University of Lyon 1, France in 1988. Since 1995, he has been in charge of the master's degree “Computer Science Engineering for Decision and Economic Evaluation” at the University of Lyon 2. He is a member of the Decision Support Databases research group within the ERIC Laboratory. His main research subjects are data warehousing, multi-dimensional databases and OLAP. His current research concerns complex data warehousing, XML warehousing, data mining-based multidimensional modelling, OLAP and data mining combining and mining metadata in RDF form.