SDWM: An Enhanced Spatial Data Warehouse Metamodel

Alfredo Cuzzocrea[1], Robson do Nascimento Fidalgo[2]

[1] ICAR-CNR & University of Calabria, 87036 Rende (CS), ITALY
cuzzocrea@si.deis.unical.it.
[2.] CIN, Federal University of Pernambuco, 50.732970 Recife (PE), BRAZIL
rdnf@cin.ufpe.br.

Abstract. Some research has been done in order to define metamodels for Spatial Data Warehouses (SDW) modeling. However, we observe that most of these works propose metamodels that mix concepts of DW modeling (i.e. dimensions and their descriptive attributes) with concepts of OLAP modeling (i.e. hierarchies and their levels). We understand that this mix is a possible limitation, because a DW (conventional or spatial) is essentially a large data repository, which can be analyzed/queried by any data analysis technology (e.g. GIS, Data Mining and OLAP). With aim of overcoming such limitation, in this paper we propose a SDW metamodel, named *Spatial Data Warehouse Metamodel* (SDWM), which describes constructors and restrictions needed to model SDW schemas. We have implemented a CASE tool according to our metamodel and, by exploiting this tool, we have designed two demonstrative SDW schemas.

Keywords: Data Models, Spatial Databases, Data Warehouse.

1 Introduction

The process of decision-making may involve the use of tools, such as Data Warehouse (DW) [6], On-Line Analytical Processing (OLAP) [11], Geographical Information Systems (GIS) [13] and Data Mining [12]. In this context, there are two different types of technologies: one for storing data (DW) and other for querying data (OLAP, GIS and Data Mining). That is, from a database, any query tool may be used to analyze its data. Hence the DW concepts must be independent of those associated to query tools, in particular, with regard to OLAP tools, which has concepts (i.e. hierarchy and its level) that are frequently mixed with the concepts of DW modeling (i.e. dimensions and its descriptive attributes). That is, in DW there are neither hierarchies nor levels, because if these concepts were intrinsic to a DW, any query tool for DW would be able to process multilevel queries, but only OLAP query tools can do it. In the next paragraphs of this Section we present the main concepts and techniques that may be used for DW/SDW modeling, our research problem and how our paper is organized.

DW is a typically-large data repository that is usually designed using the star model [6], which has two types of tables: fact and dimension. A fact table stores some metrics of a business, while a dimension table to hold its descriptive information. There are many techniques/concepts for modeling dimensions and fact tables. However, only some techniques/concepts bring additional information, required to generate correct code, namely: degenerate dimensions, role-playing dimensions and bridge tables (or many-to-many relationship). That is, a degenerate dimension ensures that it can be used only in a fact table and that it will be part of the identifier of the fact table, but it cannot be a reference/link for a dimension. In turn, a role-playing dimension allows the creation of different views of the same dimension. Finally, a bridge table (or many-to-many relationship) allows to create a third table with two one-to-many relationships, where we can specify the name of this table and some additional attributes.

A lot of data stored in a DW has some spatial context (e.g., city, state and country). This means that if one intends to properly use this data in decision support systems, it is necessary to consider the use of a Spatial DW (SDW). A SDW is an extension of the traditional DW. It has an additional spatial component (a spatial feature type) that we define from a position (a geometric attribute) more a location (a descriptive attribute), where the location is optional. Basically, a SDW extends the star model through the inclusion of this spatial component in dimensions or in fact tables. Much research has focused in SDW modeling (see Section 4). However, we identified that most of these works defines metamodels that mix concepts for dimensional modeling with concepts for cube modeling, which we disagree, because, as already stated, a DW (conventional or spatial) can be analyzed/queried by any data analysis tool (not only OLAP). With aim of overcoming such limitation, we propose the Spatial Data Warehouse Metamodel (SDWM), which is presented using UML metaclasses (see Section 2).

The remaining of this paper is organized as follows. In Section 2 we propose the SDWM Metamodel and present its definitions. Next, in Section 3 we give an overview about our CASE tool for helping in the SDW modeling tasks and present a practical application of our metamodel and CASE tool. Then, in Section 4 we make a brief discussion about some existing works for SDW metamodel/CASE tool. Finally, in Section 5 we present some conclusions and indications for future work.

2 SDWM: A Spatial Data Warehouse Metamodel

SDWM is a metamodel that embeds the following significant features: (*i*) disassociating DW modeling from OLAP cube modeling; (*ii*) representing the spatiality in a SDW simply stereotyping attributes/measures as spatial, rather than stereotyping dimension/fact table as spatial or hybrid; (*iii*) capturing whether the geometry of a spatial attribute/measure can be normalized and/or shared; (*iv*) supporting the following DW modeling techniques: degenerated dimension, many-to-many relationship (bridge table) and role-playing dimensions; (*v*) providing a set of stereotypes with pictograms that aims to be concise and friendly; (*vi*) being used as a basic metamodel for a CASE tool that aims to model logical schemas of SDW, as well as to check whether these schemas are syntactically valid. In Figure 1 we introduce SDWM using the UML class diagram.

In Figure 1, we have three enumerations, which cover one of the possible values for an attribute. The Cardinality enumeration represents whether the relationship is many-to-one, one-to-many or many-to-many. This enumeration is important to define the primary/foreign key (like in R-DBMS) or OID/REF (like in OR-DBMS). That is, the table on the "many" side has a foreign key (or a REF) to the table of the "one" side. With respect to many-to-many cardinality, we apply the bridge table technique, which creates a third table with two one-to-many relationships. In turn, the *DataType* and *GeometricType* enumerations represent the primitive or spatial data types supported by SDWM, respectively. We highlight that these enumerations are just data type indications, which will be translated for specific data types of a DBMS. Furthermore, we also point out that the spatial data types are conform to the *Simple Feature Access* (SFA) specification of *Open Geospatial Consortium* (OGC).

Our metamodel has five main metaclasses, namely: *Schema*, *Relationship*, *Table*, *DimensionColumn* and *FactColumn*. *Schema* is the root metaclass that corresponds to the drawing area for a SDW schema. For this reason, *Schema* is a composition of zero or more *Table* and zero or more *Relationship*. At last, *FactColumn* and *DimensionColumn* are just a set of different types of columns. Besides the main metaclasses, our metamodel has eight specialized metaclasses, namely: *Fact*, *Dimension*, *Bridge*, *SpatialMeasure*, *DegenerateDimension*, *ConventionalMeasure*, *SpatialAttribute* and *ConventionalAttribute*. That is, a *Table* is specialized in *Fact*, *Dimension* or *Bridge*, which capture the concepts of fact table, dimension table and a bridge table, respectively. A *FactColumn* is specialized in *SpatialMeasure*, *DegenerateDimension* and *ConventionalMeasure*, which correspond to a spatial feature type, a descriptive attribute and a measurable attribute, respectively. Finally, a *DimensionColumn* is specialized in *SpatialAttribute* and *ConventionalAttribute*, which represent a spatial feature type and a descriptive attribute. Note that, *Fact* is a composition of zero or more *FactColumn* and zero or more *ConventionalAttribute* and each *Dimension* or each *Bridge* is a composition of zero or more *DimensionColumn*. We highlight that a *Dimension* table differs from a *Bridge* table because they have different stereotypes (i.e. they represent different concepts), a *SpatialMeasure* differs from a *SpatialAttribute* because they have different stereotypes and a *SpatialAttribute* is a feature type that always has its position (or geometric information) plus its location (or descriptive information) to represent the spatiality in a SDW, while a *SpatialMeasure* may have only its geometric information, since it can be stored without its descriptive information (*hasDescription* = false). That is, on the one hand, whether a feature type is defined as a *SpatialMeasure*, it can store only its position (e.g. geometries of farms); on the other hand, whether this feature type is defined as a *SpatialAttribute*, it has to store its position and location (e.g. geometries and descriptions of farms). In turn, a *DegenerateDimension* differs from a *ConventionalAttribute*, because they have different stereotypes and only the *DegenerateDimension* can be part of fact table identifier, as well as only the *DegenerateDimension* can be defined in a fact table.

In order to capture the tables that are source and target in a relationship, we have the associations named *Source* and *Target* between *Table* and *Relationship*. The metaclass *Relationship* has *cardinality* and can have a *role* for expressing, respectively, the maximum number of instances of the relationship and a particular view of a dimension associated with a fact table (i.e. a role-playing dimension). Another important attribute is *name*. This attribute stores a label that identifies a metaclass.

In order to define whether the position of spatial measure/attribute must be normalized in a different table from its location, the *isNormalized* attribute is defined as a *Boolean*. That is, whether this attribute is defined as true, the geometric information is normalized in a separate table from the table that stores the descriptive information of the spatial attribute/measure. Otherwise (*isNormalized* = false), the geometric information is defined in the same table that stores the descriptive information of spatial attribute/measure. SDWM also allows to define whether the spatial (or geometric) information can be shared among several spatial attributes/measures. To accomplish this, it is necessary to define the same name and the same geometric type. Furthermore, for each spatial attribute/measure that will share its geometry, the attributes *isNormalized* and *isShared* must be defined as true. The default value for *isNormalized* and *isShared* is false and, in this case, there is not a special notation (i.e. the attribute/measure is written in regular font). However, when *isNormalized* or *isShared* are defined as true the attribute/measure appears in bold and/or italic font, respectively.

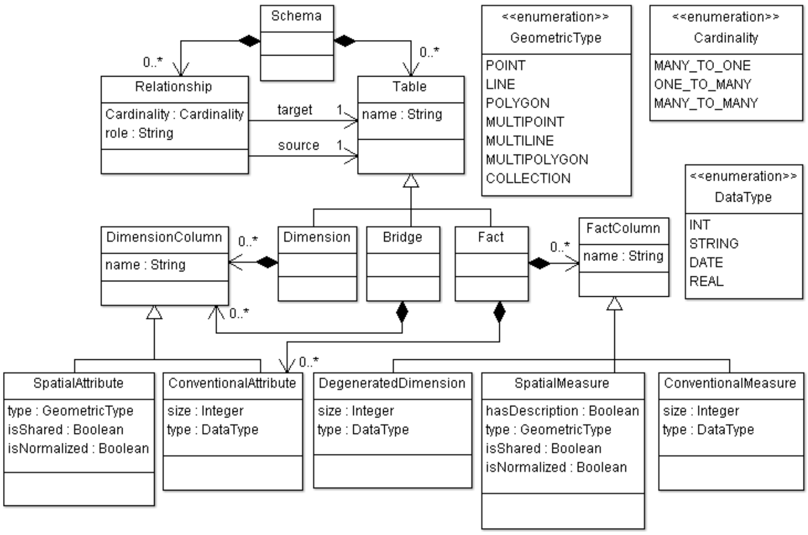
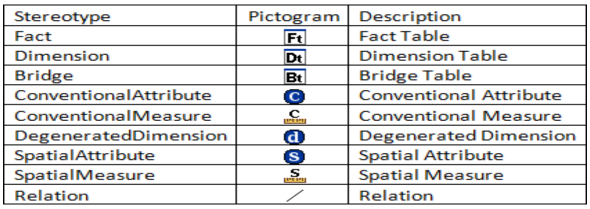


Fig. 1. SDWM Metamodel.

Spatial measures have the attribute *hasDescription*, which allows to define whether the spatial measure has a description (*hasDescription* = true) and a geometry or, otherwise (*hasDescription* = false), whether the spatial measure has only a geometry. *DegeneratedDimension, ConventionalMeasure* and *ConventionalAttribute* may have a size and each specialization of *DimensionColumn* and *FactColumn* has an associated type from our allowed data types (i.e. *DataType* and *GeometricType* enumerations). SDWM uses stereotypes and pictograms to increase its expressiveness (see Figure 2) and to represent primitive types and spatial types (see Figure 3).

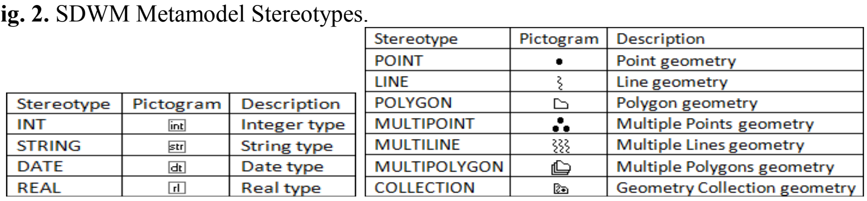
3 A Real-Life SDW

In order to evaluate the correctness and usefulness of our metamodel, we developed a CASE tool, called SDWCASE, that was used to design a SDW with meteorological data from the *Laboratory of Meteorology of Pernambuco* (LAMEPE). This laboratory has a net of meteorological *Data Collection Platform* (DCP) for monitoring atmospheric conditions. SDWCASE is a CASE tool that offers a concise and friendly GUI that is based on the set of stereotypes with pictograms presented in Figures 2 and 3. With our CASE tool, the designer can interact with the SDW schema by inserting, excluding, editing, visualizing at different zoom levels, exporting a figure (e.g. JPG, GIF, PNG) or XMI (*XML Metadata Interchange*) file. Moreover, SDWCASE also allows the validation of the modeled schema. For example, (*i*) two tables (dimension or fact) or two attributes (in the same table) cannot have the same name; (*ii*) a table cannot be associated with itself; (*iii*) measures and degenerated dimensions can only exist in a fact table; (*iv*) dimension tables and bridged tables can only have attributes. The first and the second validations are ensured by programming, but the third and the fourth are intrinsically/automatically ensured by our metamodel (see Figure 1). SDWCASE is implemented in Java using the *Eclipse Graphical Modeling Framework* (GMF) plus the *Eclipse Modeling Framework* (EMF) and, in its current version, it generates code only for PostgreSQL with PostGIS. However, it can be done for any spatial DBMS.



Stereotype	Pictogram	Description
Fact	Ft	Fact Table
Dimension	Dt	Dimension Table
Bridge	Bt	Bridge Table
ConventionalAttribute	C	Conventional Attribute
ConventionalMeasure	C	Conventional Measure
DegeneratedDimension	d	Degenerated Dimension
SpatialAttribute	S	Spatial Attribute
SpatialMeasure	S	Spatial Measure
Relation	/	Relation

Fig. 2. SDWM Metamodel Stereotypes.

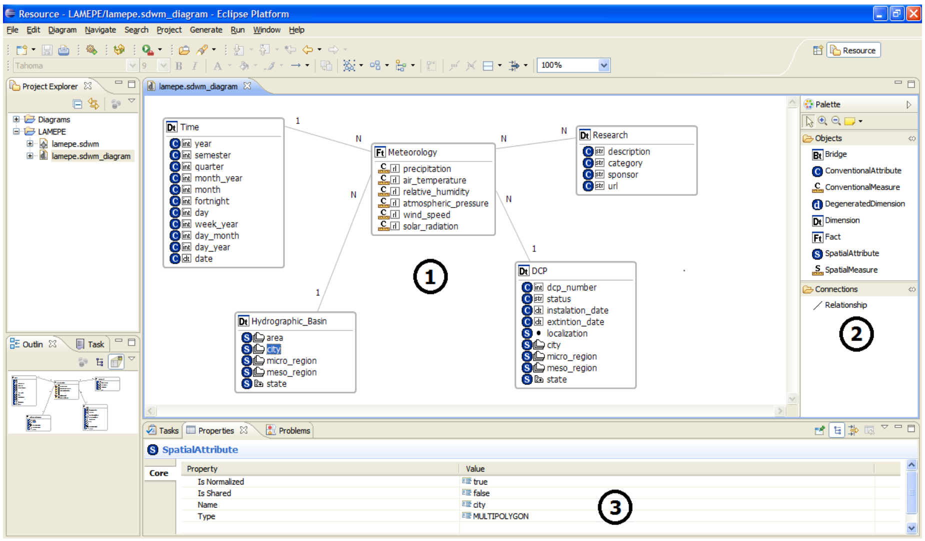


Stereotype	Pictogram	Description
INT	int	Integer type
STRING	str	String type
DATE	dt	Date type
REAL	rl	Real type

Stereotype	Pictogram	Description
POINT	●	Point geometry
LINE	⌇	Line geometry
POLYGON	⏢	Polygon geometry
MULTIPOINT	⁖	Multiple Points geometry
MULTILINE	⌇⌇⌇	Multiple Lines geometry
MULTIPOLYGON	⏢	Multiple Polygons geometry
COLLECTION	⏢	Geometry Collection geometry

Fig. 3. SDWM Primitive Type Stereotypes and Geometry Type Stereotypes.

In Figure 4 we show the SDWCASE GUI with the LAMEPE SDW using many-to-many relationship. The SDWCASE GUI has a palette (area 2 in Figure 4) with all elements (defined in SDWM) that the designer needs to model a SDW. The modeling tasks starts with a click on the desired element in the palette and place it in the drawing area (area 1 in Figure 4). Next, the designer may edit the properties of the element (area 3 in Figure 4), and add new elements or relationships. Note that (*i*) each element is easily identified by its pictogram and (*ii*) the SDW schema is concise. That is, only using spatial attributes/measures, we can represent the spatiality in a SDW with a short notation. In Figure 4, we have one fact table, four dimension tables, and conventional and spatial attributes, which are stereotyped according to SDWM pictogram. In this figure you can note that there is a many-to-many relationship between the fact table *Meteorology* and the dimension *Research*. In this case, our CASE tool abstracts the creation of a third table to implement this relationship. However, an explicit bridge table also can be defined in SDWCASE. Another schema using bridge table, role-playing dimensions, spatial measure, degenerated dimension and conventional attribute can be seen in [2].



Property	Value
Is Normalized	true
Is Shared	false
Name	city
Type	MULTIPOLYGON

Fig. 4. SDWCASE GUI with LAMEPE SDW using many-to-many relationship.

4 Related Work

In order to do a systematic evaluation of these works, we are using the following features to compare, which we retain critical for modeling SDW:

1. disassociating DW modeling from OLAP Cube modeling;
2. making a CASE tool available to users;
3. supporting the following DW modeling techniques: degenerated dimensions, many-to-many relationships and role-playing dimensions;
4. supporting spatial attributes rather than spatial or hybrid dimension;
5. supporting spatial measures;
6. providing a set of stereotypes with pictograms that aim to be concise – i.e., it provides a short notation;
7. capturing whether the geometry of a spatial attribute/measure can be normalized and/or shared.

Bédard et al. [1, 9] define three types of spatial dimensions: the non-geometric spatial dimensions (all level are conventional data), the geometric spatial dimensions (all level are spatial data) and the mixed spatial dimensions (it has conventional and spatial data in the same dimension). The authors also differentiate numerical and spatial measures, where the spatial measures are considered as a collection of geometries.

Fidalgo et al. [3, 10] define a metamodel and a CASE tool for SDW modeling. Similarly the previous work, the authors specify concepts of measures (conventional or spatial) and dimensions (conventional, spatial or hybrid). Moreover, the metamodel and the CASE tool provide a set of stereotypes and pictograms, which are for SDW modeling.

However, both, metamodel and CASE tool, although support the technique degenerated dimension, they do not support many-to-many relationships neither role playing dimensions techniques nor spatial attributes.

Malinowski and Zimányi [7, 8] define an extension of ER model to represent dimensions, hierarchies and spatial measures/levels. The extension makes use of classes and relationships, both stereotyped with spatial pictograms, to model the geometry of spatial levels and the topological relationships between these levels.

Glorio and Trujillo [4, 5] define an UML profile and a CASE tool that use a set of stereotypes and pictograms for dimensions, hierarchies and spatial measures/levels. Although this work supports the technique degenerated dimension, it does not support many-to-many relationships neither role playing dimensions.

In short, all works support spatial measure, but no work supports spatial attributes. Consequently, no work captures whether the geometry of a spatial attribute must be normalized and/or shared. Moreover, only Fidalgo et al. [3, 10] do not mix DW modeling with OLAP modeling, as well as, only Fidalgo et al. [3, 10] and Glorio and Trujillo [4, 5] support degenerated dimension technique, but these works do not support many-to-many relationships neither role playing dimensions techniques. Finally, although most of these works provides a set of spatial stereotypes with pictograms, these works represent the spatial information as a stereotyped class, which does not provide a concise/short notation, because it pollutes the SDW schema whether it has much spatial information. In Table 1 we compare our work with the related works discussed here.

Table 1. Analysis of related works and our proposal.

	Bédard et al.	**Fidalgo et al**	**Malinowski and Zimányi**	**Glorio and Trujillo**	**Our Proposal**
DW vs. OLAP Modeling	NO	YES	NO	NO	YES
CASE Tool	NO	YES	NO	YES	YES
Degenerated Dimensions	NO	YES	NO	YES	YES
M-N Relationships	NO	NO	NO	NO	YES
Role-Playing Dimensions	NO	NO	NO	NO	YES
Spatial Attributes	NO	NO	NO	NO	YES
Spatial Measures	YES	YES	YES	YES	YES
Short notation	NO	NO	NO	NO	YES
Normalized/Shared Geo.	NO	NO	NO	NO	YES

5 Final Remarks

Many proposals have focused in metamodel and/or CASE tool for SDW. However, most of these works defines metamodels that (*i*) mix concepts of DW modeling with concepts of the OLAP modeling; (*ii*) does not support important techniques of DW modeling, (*iii*) represents the spatiality in a SDW stereotyping the dimensions and fact table as spatial or hybrid, rather than stereotyping the attributes/measures as spatial; (*iv*) defines a complex taxonomy of spatial dimensions and measures, (*v*) does not provide a concise and friendly set of stereotypes with pictograms; and/or (*vi*) is not used as a basic metamodel for a CASE tool. In order to give a contribution to solve the previous problems, we have proposed the *Spatial Data Warehouse Metamodel* (SDWM), which defines the constructors and the restrictions needed to design SDW schemas that are consistent and unambiguous. Our metamodel is more straightforward and more expressive than its related works, because it (*i*) represents the spatiality in a SDW assigning spatial stereotypes in attributes and measures, (*ii*) disassociates the DW modeling from the OLAP cube modeling, (*iii*) captures whether the geometry of a spatial attribute/measure can be normalized and/or shared, (*iv*) proposes a set of stereotypes with pictograms that aims to provide a short/concise notation, and (*iv*) supports the following DW modeling techniques: degenerated dimension, bridge table and role-playing dimensions. For this, SDWM can be used as a basic metamodel for a CASE tool that aims to make the design of invalid SDW schema much harder, as well as to make the automatic SQL/DDL code generation from these schemas.

To evaluate our proposal, SDWM has been implemented in a CASE tool and tested with a case study that illustrates a use of our metamodel/CASE tool, demonstrating that the semantic and syntax of our metamodel are modeled correctly, and its notation is unambiguous. The CASE tool is named SDWCASE. It is implemented in Java and in its current version, generates SQL/DDL code for PostgreSQL/PostGIS. In future work, other spatial DBMS will also be covered. Other direction for future work is the: definition of a metamodel and CASE tool to model and query a Spatial OLAP cube.

Acknowledgments This work was partially supported by the process BEX 5026/11 CAPES/DGU from the CAPES Brazilian foundation.

References

[1] Bédard, Y., Merrett, T. and Han, J.: Fundamentals of spatial data warehousing for geographic knowledge discovery. In: Geographic Data Mining and Knowledge Discovery 2001: 53-73.

[2] Del Aguila, P. S. R., Fidalgo, R. N., Mota, A.: Towards a more straightforward and more expressive metamodel for SDW modeling. DOLAP 2011: 31-36

[3] Fidalgo, R. N., Times, V. C., Silva, J. Souza, F. F.: GeoDWFrame: A Framework for Guiding the Design of Geographical Dimensional Schemas. In: DaWaK 2004: 26-37.

[4] Glorio, O., and Trujillo, J.: An MDA Approach for the Development of Spatial Data Warehouses. In: DaWaK 2008: 23-32.

[5] Glorio, O., and Trujillo, J.: Designing Data Warehouses for Geographic OLAP Querying by Using MDA. In: ICCSA (1) 2009: 505-519.

[6] Kimball, R. and Ross, M.: The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, second ed., Wiley, NewYork, 2002.

[7] Malinowski, E. and Zimányi, E.: Representing spatiality in a conceptual multidimensional model. In: GIS 2004: 12-22.

[8] Malinowski, E., and Zimányi, E.: Advanced Data Warehouse Design From Conventional to Spatial and Temporal Applications, Springer, 1st ed. 2008.

[9] Pestana, G., Silva, M. M. da, & Bédard, Y.: Spatial OLAP modeling: an overview base on spatial objects changing over time. In ICCC 2005: 149-154.

[10] Silva, J., Oliveira, A. G., Fidalgo, R. N., Salgado, A. C. and Times, V. C.: Modelling and querying geographical data warehouses. In: Inf. Syst. 2010 35(5): 592-614

[11] Thomsen, E.: OLAP Solutions: Building Multidimensional Information Systems. John Wiley and Sons, 2nd ed. 2002

[12] Witten, I. H. and Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann , 2nd ed. 2005.

[13] Worboys, M., and Duckham, M. 2004. GIS: A Computing Perspective, Taylor & Francis. 2nd ed. 2004.