Modelling and Manipulating Multiple Representations of Spatial Data

Christelle Vangenot[1], Christine Parent[2], Stefano Spaccapietra[1]

Swiss Federal Institute of Technology, Database Laboratory, Lausanne, Switzerland, christelle.vangenot@epfl.ch, stefano.spaccapietra@epfl.ch[1]
University of Lausanne, INFORGE, Lausanne, Switzerland, christine.parent@hec.unil.ch[2]

Abstract

One of the requirements which is poorly supported by spatial data models is a consistent management of different representations of the same spatial phenomena from different viewpoints or at different resolutions. This need is well known by users and designers. Modelling of applications, where users share the same database for different contexts and cartographic applications are examples of environments where such a need arises. This paper proposes a conceptual data model providing full support for multiple representations of the same real world data. The model addresses two complementary aspects: the integrated approach, that leads to the definition of customised database items, and the inter-relationship approach, where the representations are linked through inter-representation links. Both approaches use the stamping technique to differentiate among multiple representations of a given phenomenon and to access a particular representation. Finally, we focus on consequences of multiple coexisting representations on data manipulation. This proposal has been tested and validated with users, and implemented as a front-end to existing DBMS[1].
Keywords: multi-representation, multi-resolution, spatial data modelling, databases.

---

[1] We gratefully acknowledge support for this research by the European Economic Community (MurMur Project - "Multiple Representation – Multiple Resolution" - IST 10723), by the Swiss OFES (grant 99999), by EEC Interreg II Program (Cobalt Project), and by the Swiss National Research Fund (grants 8765-2000 & 112-2002).

1 Introduction

Interoperability supports communication and access among information repositories, giving users an opportunity to locate complementary information about the same or related facts from various sources that have been independently developed. Unfortunately, semantic interoperability is not easy to achieve, as related knowledge is most likely to be described in different terms, using different assumptions and different data structures. Reconciling this heterogeneity to build a fully integrated database is a complex problem and largely unresolved.. A simpler step is the identification of related knowledge and the provision of a mechanism that in some way establishes the relationships among different representations of the same elements, i.e. that supports multiple representations of data. One well known example of applications where multiple representations is a crucial need and are not supported by current technology is cartographic applications. Map producers often need to build maps of the same geographic region at different levels of detail. Cartographic generalisation, the process of deriving a less detailed map from a more detailed one, is a complex process that usually remains partially interactive (Müller *et al.* 1995). Often, map producers have to maintain one database per scale, and do not directly maintain interrelationship between multiple scale databases. As a result, there is generally no update propagation and inter-database consistency is lost. A multiple representation GIS would allow storing all representations in a single database, and support the consistency of the different representations through appropriate and automatic update propagation. Beyond cartography, multiple representations of geographic data are needed to be able to serve multi-disciplinary user communities, as the same piece of land may support analysis, planning, and forecast activities by city administrations, environmentalists, sociologists, botanists, zoologists, among others.

Research on multi-representation has often been addressed in specific contexts such as multi-scale databases, views, and multi-instantiation, each of which are separate research areas with specific requirements. A significant part of the research in multi-scale databases was inspired by the largely hierarchical nature of transitions between scales (Weibel *et al.* 1999). (Timpf 1998) proposed to keep the representations of maps at different scales in hierarchical data structures, where levels correspond to increasing detail and the representations of the same object are linked together. In (Kilpelaïnen 1998), these multiple representation links are used to propagate updates. (Martel 1999) proposes patterns to deal with multi-representation problems. His solution, implemented in VUEL (Bernier *et al.* 2001) allows the association of several thematic, graphic and semantic characteristics to the same geographic object. (Stell *et al.* 1998) see the database organised as a stratified map space with transformation functions, where each map gathers objects that share the same semantic and spatial granularity. (Devogele *et al.* 1998) built a federated database on top of mono-scale databases. Geodyssey (Jones *et al.* 1996), a deductive knowledge-based system allowing multi-scale and temporal representations, incorporates reasoning processes to maintain consistency and propagate updates.

This paper presents a set of modelling concepts allowing each information fact to be described through multiple, consistent, and possibly irreducible representations. The next section defines which aspects of multiple representations we deal with. Section 3 presents our suggested framework and data structures that adequately support multi-representation. Sections 4 and 5 detail the concepts and design principles for multiple representation modelling. Section 6 suggests some extensions to algebraic operators to be able to manipulate our new modelling concepts. Section 7 presents the implementation of the model. The conclusion points at additional work that complements the results reported in this paper.

2 What is Multiple Representation?

Databases are intended to keep an integrated and consistent set of data that provides the information needed to support application requirements from one or several user communities. These data represent real-word phenomena that are of interest to users of the database. While the real world is assumed unique, the way it is represented depends on the intended use. Thus, different applications that share interest in the same real-word phenomena may have different perceptions and therefore require different representations; i.e. different sets of objects, links, properties, and/or different values. Their needs differ because they don't share the same viewpoint, they don't need data acquired at the same instant or with the same cartographic resolution. Multiple resolutions may be used in each of these cases as well as in others. In this paper, we explicitly deal with two of them, but our solution is generic and can be applied to other aspects. We selected the viewpoint aspect, which is fundamental for data sharing, and we address the aspect concerning the level of detail (or resolution) of the geographic data.

A **viewpoint** is the expression by a group of users about their specific interests in data management. It acts as a complex abstraction process where objects and links are filtered, leaving out whatever in the real world is not of interest for the particular viewpoint. This delimits the so-called universe of discourse. Then each phenomenon in the universe of discourse is described according to the viewpoint. This induces multiple design decisions, such as the choice of a representation concept (the same phenomenon may be represented as an object, an attribute or a relationship), the elaboration of its type (among all possible properties of the phenomenon, only a subset is of interest for the viewpoint), and for each property the choice of its representation.

Every representation conveys a simplified description of the reality. This materialises into a level of detail that characterises both spatial and thematic characteristics of information (Molenaar 1998). In the spatial dimension, the level of detail, called **spatial resolution**, determines the geometric aspects of the phenomena. Spatial resolution is defined by thresholds (Peng 1997). For instance, spatial resolution could be defined as a process of filtering out all 2D area objects whose surface is less than 100 square meters and all 1D linear objects whose total length is less than 150 meters. Beyond filtering, spatial resolution also has a smoothing effect: given a detailed geometry, a less-detailed representation will retain a simplified geometry that leaves out all irregularities whose size is less than the given threshold. It also has a merging effect: distinct geometries whose distance is less than the threshold, collapse into a unique geometry. A set of buildings, for instance, may collapse to form a single built-up area when the details on individual buildings are no longer of interest. The concept of multiple levels of detail also applies to thematic data, and is called **semantic resolution.** Semantic resolution allows filtering out objects/relationships/attributes that are not relevant while working at a specific level of detail. Semantic resolution may be specified by the number of classes, the membership rules of classes, the number of attributes of classes, the granularity of hierarchical value domains (Rigaux *et al.*1995), the depth of is-a and aggregation hierarchies. In the following sections, we use the term resolution to cover both spatial and semantic resolution.

3 The Multi-Representation Framework

3.1 MADS Data Model

Our objective is to define a set of concepts necessary for describing datasets with multi-representation. Our prototype was built as an extension to an already existing data model, the MADS data model (Parent *et al.* 1998). MADS is an object+relationship, spatio-temporal, conceptual data model. In this model, we assume that the real world of interest that is to be represented in the database is composed of objects, their links, and their static and dynamic properties (attributes and methods). Spatial and temporal aspects may be associated at the different structural levels: object, attribute and relationship. The spatial characteristics of an object convey information about its location and its extent, while its temporal criteria describe its lifecycle. Attributes may have spatial (e.g. point or area) or temporal (e.g. instant or time interval) domains of values. They also may be space- or time-varying. Specific relationships describe topological constraints between spatial and temporal objects. In MADS icons express information in an unambiguous and visual way. An example of a schema, drawn with the MADS schema editor, is presented in Fig. 7.

3.2 Proposed Framework

The two facets of multi-representation, viewpoint and resolution are introduced in the data model through the definition of representation stamps. A stamp is a (viewpoint-value, resolution-value) pair, for instance ("cartography", 1 meter). Stamps characterise the different representations of the real world objects. They are like meta-data specifying the context of elaboration (here the viewpoint and the resolution) of each representation. Any schema or database element, i.e. object type, relationship type, attributes, instance, value, holds one or more stamps. An

element bears several stamps when it is shared by several contexts. For instance, Fig. 1. Stamping attributes and their values
in which the object type Road holds two stamps, e1 and e2 (for ("cartography", 1) and ("hydrography", 5)), which means that this object type is shared by the contexts defined by the stamps e1 and e2. Beyond modelling, stamps are also used during data manipulation for filtering or directing the access to data. In our model, users' transactions also bear stamps (see section 6) that specify which data the transaction may access.

We propose two complementary approaches to define multiple representations for the same real world phenomenon. We denote the first strategy as the **integrated approach**. The idea is to have a unique name and container for various representations while identifying the specifics of each representation using the stamping technique. Each construct of the data model can be turned into a multi-representation construct by stamping it with a representation stamp. A multi-representation construct (object, attribute, link) embeds within an integrated definition the different representations, each representation being characterised by one or several stamps. Section 4 below discusses multi-representation object types, attributes, and relationship types.

When representational needs for the same phenomena are so diverse that they can hardly be integrated into a common definitional framework, it may be more convenient to independently define the needed representations, and to interrelate them through dedicated links. The latter allow an expression of the multi-representation semantics. We call this the **inter-relationship approach**. These links, named correspondence relationship types in section 5, relate two representations of the same real-world entity described by different instances and pertaining to different viewpoints and/or resolutions.

The two approaches are complementary and both essential to describe multiple representations. A model offering only the integrated approach does not allow the description of correspondences between groups of objects. This may prevent the model to be applicable in situations where the multi-representation database has to be built on top of (or derived from) multiple existing data sets. Conversely, a data model with only correspondence links does not provide ‘orthogonal modelling’ tasks, as links can only be defined between objects but not between attributes or values. This means that differences in viewpoint or resolution would invoke the modelling of the related representations as objects but not as attributes or values. The designer’s freedom in choosing a modelling construct would be restricted.

4 Multi-Representation Database Elements

4.1 Stamping Object Types, Attributes and Their Values

Let us use an example to illustrate the definition of multi-representation database elements. Consider two representations of roads, identified by stamps e1 and e2. Roads are spatial objects whose spatial extent, depending on the representation, is represented as an area (most precise representation, stamp e1) or a line (less precise one, stamp e2). Moreover the road number, the name of the road, its administrative classification, its type, and the name of its main manager are interests of representation e1. Representation e2 also needs the road number, its name (but with another naming scheme), its type (but with another classification), the department in charge of it, and its set of managers. Fig. 1 shows the stamped object type Road and its attributes in which both representations (e1 and e2) are embedded. The stamps of the object type are shown in the rectangle between the rectangles containing the object type name and the list of its properties. Stamps are also written next to each attribute definition, thus specifying the stamps for which the definition is pertinent. The Road object type is stamped with two stamps, e1 and e2; that means that this object type is shared by the applications identified by the stamps e1 and e2. It can be accessed by the transactions having the stamp e1, or the stamp e2, or both stamps. Its attributes are also stamped, i.e. the structure (in term of sets of attributes) of the object type changes according to the stamps. It behaves as if it has two definitions, one for stamp e1 and one for stamp e2.

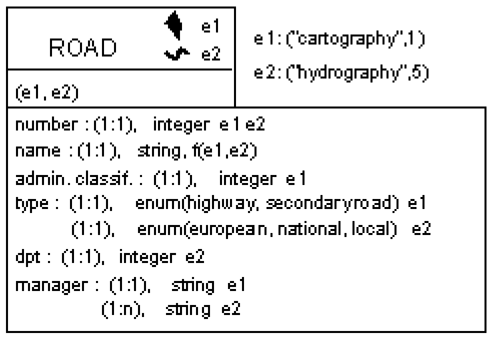

Fig. 1. Stamping attributes and their values

Furthermore, if we consider its attributes:

- The attribute *number* has the same stamps as its object type Road: This attribute is shared by both representations. It is accessible by the transactions having the stamp e1, or the stamp e2, or both.
- The annotation ‘f’(e1, e2) after the attribute *name* means that its value is varying according to stamps. It is thus possible to store several values for this attribute, one per specified stamp (or less if several stamps share a value). It is thus possible to store two values for name, like for instance, “RN7” for stamp e1 and “Route Napoléon” for stamp e2.
- The attribute *administrative classification* (respectively *dpt*) belongs to the representation e1 (resp. e2). It is only accessible by the transactions having stamp e1 (respectively e2) or both stamps.
- The attributes *geometry*, *type* and *manager*, have two definitions and therefore two values, one for the stamp e1 and one for the stamp e2. The stored values of the attributes geometry and type pertain to different value domains. It is thus possible to store the geometry of an area for stamp e1 and a line for stamp e2.

The values of the attribute manager belong to the same domain but the cardinality of this attribute is different. A single value is stored for transactions of stamp e1 and several values are stored for transactions of stamp e2.

Instances of object types are also stamped (so are instances of relationship types). They can have either the same set of stamps as their type or only a subset of it. A stamped instance has values for the attributes that are defined for its stamps. An instance of Road with the stamps (e1, e2) has values for all the attributes. Transactions of stamp e1, or e2, or (e1, e2) can see this instance: transactions of stamp e1 with the attributes defined for e1 or (e1, e2); transactions having both stamps with all attributes. An instance of Road with the stamp e1 has values for the attributes belonging to the definition at stamp e1. Transactions of stamp e1 or (e1, e2) can view this particular instance.

All the database items are stamped, however, as they are not independent, they can not be stamped irrespective of each other. We have defined rules specifying stamping consistency. The first rule says that the stamps of database elements must be included in the set of stamps of the element they are part of: Stamps of object/relationship types have to be included in the set of stamps of the schema they belong to, stamps of attributes in the stamps of the type they are defined in, stamps of attribute values in the stamps of the attribute, and stamps of instances in the stamps of their type. For instance, the attributes of the road object type may be stamped with (e1, e2) link number, or just with e1, or just with e2, link dpt.

4.2 Stamping linked elements

Accessing a relationship type is different from accessing an object type. Indeed, relationship types cannot be considered regardless of the object types they link: access to a relationship type instance must always enclose access to the object type instances linked. Hence, we rule that a transaction may only access a relationship type if its stamps also give access to all the linked object types. By default relationship types are not stamped. With respect to object types, it is possible to define multi-representation relationship types. Stamping a relationship type allows one to characterise the properties and instances of the type and to add access conditions. For example, stamping the instances of the topological relationship type ‘near’ shown in Fig. 2 allows one to describe cartographic representations at different scales, where the same house is adjacent to a road in the less detailed representation (e.g. stamp e1) and no longer adjacent in the most detailed representation (e.g. stamp e3) (Jen 1995). To describe this situation, the instance of the near relationship linking this house and this road has to be stamped with e1 only.

Fig. 2. Stamping a topological relationship type

If the relationship type is explicitly stamped, the requesting transaction must have at least one of the stamps of the relationship type, in addition to stamps giving access to the linked object types.

Topological and synchronisation relationship types are a particular kind of relationships because they invoke spatial/temporal constraints between the linked objects. Thus, transactions have to make sure they have the stamps of the geometry/ lifecycle of the linked object types.

5 Correspondence Relationship Types

Correspondence relationships allow designers to relate different representations of the same real-world entity when described by different instances. This approach allows users to deal with more situations of multiple representations. Indeed the correspondence relationship types can be of different kinds and link one or more instances. In this approach, we use the following correspondence links:

The *identity* relationship □ type is used to link instances that are alternative representations of the same real world entity, like for instance an object that is represented as a building by some users and as an historical monument by others (see Fig. 3). The two linked objects have the same object identifier. This relationship shares its semantics with the is-a link but it is more generic as it can have properties describing the link (e.g. the attribute matching-date) or useful for the propagation of the updates of one object to the other one.

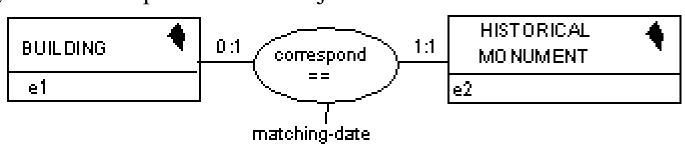

Fig. 3. Identity relationship type between two object types

The *aggregation* relationship type may be used as a correspondence relationship. The aggregation describes the situation where a real world phenomenon is decomposed in a representation. For instance in Fig. 4, a university is seen as a simple object University from one viewpoint and from another viewpoint as a set of component objects: the buildings and playgrounds of the university.

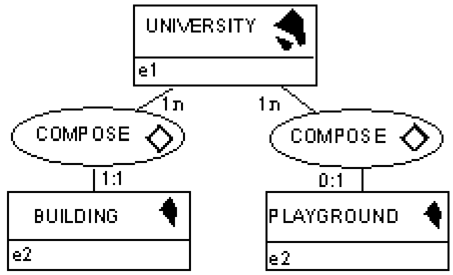

Fig. 4. Example of aggregations

The *SetToSet* relationship type is a new kind of relationship type that conveys the semantics that a group of objects corresponds to another group of objects. This correspondence cannot be captured by traditional relationships that only allow users to link one instance of each participating object type. The "set to set" correspondence is usual when describing data coming from cartographic generalisation. Indeed objects on a map do not necessarily match 1:1 with the real objects. For instance a set of aligned buildings in a built up area may be represented in a map by a set of building icons whose spatial configuration, shape, and spacing looks similar to their real world presentation.

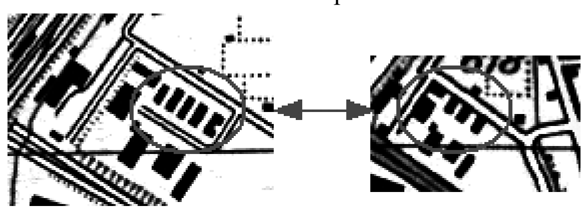

Fig. 5. An example of a SetToSet correspondence.

Fig. 5 shows an example for the same area, in which the more detailed map on the left has five buildings, while the less detailed map on the right includes only three buildings. Fig. 6 is a possible schema of the example of Fig. 5. Note that as each instance of a SetToSet relationship may link several instances of each participating object type; each role has an additional cardinality specification defining how many object instances the role may link to one relationship instance.

Fig. 6. An example of the SetToSet relationship

As for relationship types, in order to get access to a correspondence relationship a query must have at least one stamp of each of the two linked objects.

6 Data Manipulation

This section discusses the requirements that characterise the manipulation of data structures with multi-representation. To avoid ambiguities in the manipulation of multi-representation structures, queries have to convey the specification about which representations are to be used, i.e. to use stamps. The combination of the stamps specified by the query and the stamps that characterise data and metadata in the database defines which data is visible.

The first step when manipulating our multi-representation database is to specify the subset of the database that will be used. This is done with the OPEN SCHEMA command in which a set of stamps is specified. This command allows transactions to access all object types, relationship types, and attributes whose set of stamps intersect the OPEN SCHEMA set of stamps. Access to a relationship requires also that the stamps held by the transaction give access to all the linked object types. For instance, OPEN SCHEMA (My Schema,{e1}) selects the sub-schema composed of object types and attributes that have the stamp e1 and relationship types and link object types with (at least) stamp e1. Access to the object type Road of Fig.1 would be limited to the following attributes: number, name with the value of stamp e1, admin. classif., type, manager, and geometry with the e1 definition.

The schema selected with the OPEN SCHEMA command is not necessarily a mono- representation schema. Thus, queries are likely to manipulate stamped representations. To unambiguously manipulate stamped data, they also have to contain stamps. Those stamps make up additional selection criteria when accessing data. Queries within a transaction by default inherit stamps from the transaction. Consider, for instance, a query on the schema of Fig. 1 to retrieve Road instances where number=7. The query expressed with the MADS algebra is:

SELECTION [number = 7] Road

The query is automatically complemented with the stamps associated to the transaction. There is no stamp specification for the attribute number as there is no ambiguity accessing it. If the user wants to limit his/her query to the roads of stamp e1, he/she needs to complete it as:

SELECTION [number = 7] (Rstamp=e1) Road

This query gives access to the definition and instances of the Road type that holds for e1. The schema of the result is composed of attributes from Road whose stamps include e1. The selected instances are those whose number is 7 and that bear at least the stamp e1. As presented above, an attribute may hold several values and/or definitions. To access such attributes, the query also has to specify unambiguously which value or definition is required. For instance, the following query:

SELECTION [(R-stamp=e1) type = "highway"] Road returns the roads whose attribute type has the value "highway" for its definition at stamp e1.

SELECTION [name = (R-stamp=e1) "Beaulieu"] Road

returns the roads whose attribute name has the value "Beaulieu" at stamp e1.

Due to multi-representation types, two operations are needed to insert instances in the database. The first one inserts a new instance in a type. Users have to provide a value for all the mandatory attributes corresponding to the format at the given stamp. For instance, the following expression adds a new instance of Road that is stamped with e1:

INSERT INTO ROAD VALUES (stamp = {e1} ,
geometry = area {(x1,y1), …(xn,yn)} , number = 7, name = "RN7" ,
admin.classif. = 5, type = “secondary road”, manager = “Dupont”)

Another operation is needed to add a new representation to an instance that was previously defined. For instance, to add a new representation to the RN7 Road instance the following expression reads:

ADDREP TO ROAD WHERE number = 7 VALUES (stamp = {e2} ,
name = “Route Napoléon” , type = “national”, dpt = 21,
manager = {“Dupont”, “Durant”, “Rochat”})

7 Implementation and Tests

The results presented in this paper have been implemented and tested as part of the work done in the European project MurMur [MUR00]. The goal of the project is to develop, test and validate a software layer implemented on top of the Oracle 8i DBMS. This layer should allow users to:

- Define a spatio-temporal database schema using a multiple representations data model,
- Store their data in the underlying database (Oracle 8i) in a totally transparent way (translations to the model of Oracle are done by the Murmur software),
- Formulate and run interactive queries on the database.

The schema of Fig. 7 shows a screen copy of a multi-representation schema defined with the MurMur schema editor. Stamps are visualised by different colours (in black and gray in the figure). A test version of the prototype can be downloaded at http://lbd.epfl.ch.

The proposed data model and tools have been tested and validated on two real case studies. The first one is a cartographic application from IGN, the French national mapping agency, and involves three existing databases describing the French territory at different scales. The second one is a risk management application from the Cemagref research centre, and involves integrating different thematic databases with temporal data.

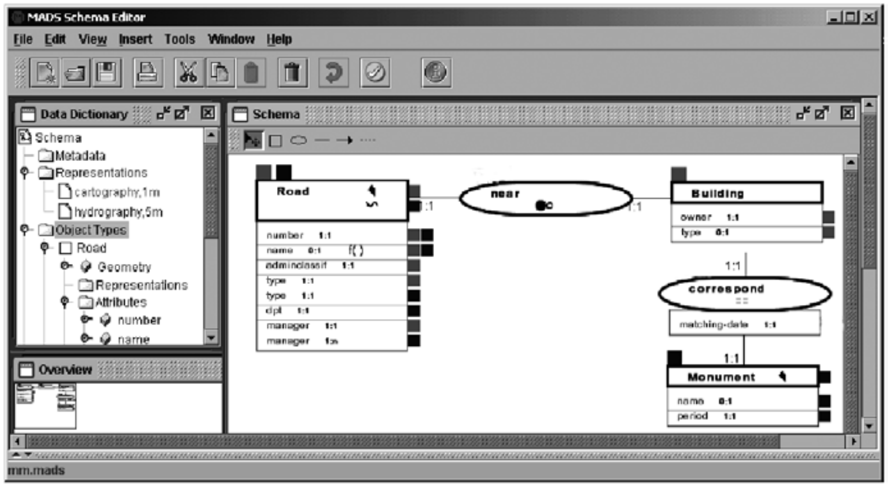

Fig. 7. Screen copy of a session of the MurMur schema editor

8 Conclusion

In this paper we propose a set of concepts as an extension of the MADS data model to support multiple representation of data. Representations may vary according to different criteria, the viewpoint, that is the materialisation of user's needs, and the resolution, that specifies the level of detail of a representation. We propose two approaches to describe multiple representations of real world objects: the integrated approach where the stamping technique is used to define customised data structures and the inter-relationship approach where representations are linked through correspondence links. MADS operators are then extended to be able to manipulate stamped data and metadata. More information about modelling and manipulation facilities is available in (Vangenot 2001). Work is now underway to address problems of consistency between representations. Our goal is to deal with consistency between data of different resolution or viewpoint in the database system (Egenhofer *et al.* 1994), and to find generic rules valid for whatever the application.

Further, our objective is to propose a language and mechanisms allowing designers to define application specific rules.

References

Bernier E, Bédard Y, Devillers R (2001) Automatic generalization and multiple representation for Spatial OLAP. In: Proceedings Geo Information Fusion and Revision. Laval University, Quebec, Canada

Devogele T, Parent C, Spaccapietra S (1998) On Spatial Database Integration. International Journal of Geographical Information Systems 12(4): 335-352

Egenhofer MJ, Clementini E, Di Felice P (1994) Evaluating inconsistencies among multiple representations. In: Proceedings of the Sixth International Symposium on Spatial Data Handling. Edinburgh, Scotland, pp 901-920

Jen TY (1995) Evolution of spatial relationships for interoperability of heterogeneous spatial databases. In: COSIT'95 Doctoral Consortium, Geoinfo-Series. Department of Geoinformation of Technical University, Vienna, pp 58-66

Jones CB, Kidner DB, Luo LQ, Bundy GL, Ware JM (1996) Database design for a multi-scale spatial information system. International Journal of Geographical Information Systems 10(8): 901-920

Kilpelaïnen T (1998) Maintenance of topographic data by multiple representations. In: Proceedings for the Annual Conference and Exposition of GIS/LIS '98. Forth Worth, Texas

Martel C (1999) Développement d'un cadre théorique pour la gestion des représentations multiples dans les bases de données spatiales. M.Sc. thesis, Laval University, Canada

Molenaar M (1998) Composite objects and multiscale approaches. In: An introduction to the theory of spatial object modeling for GIS. Taylor & Francis, pp 161-191

Müller JC, Lagrange JP, Weibel R, Salgé F (1995) Generalization: State of the art and issues. In: Müller JC, Lagrange JP, and Weibel R (eds) GIS and Generalization: Methodology and Practice. pp 3-17

MurMur Consortium (2000) Supporting Multiple Representations in Spatio-Temporal databases [online]. In: Proceedings of the 6th EC-GI & GIS Workshop. Lyon, France. Available from: http://lbd.epfl.ch/e/MurMur

Parent C, Spaccapietra S, Zimanyi E, Donini P, Plazanet C, Vangenot C (1998) Modeling Spatial Data in the MADS Conceptual Model. In: Proceedings of 8th International Symposium on Spatial Data Handling. Vancouver, Canada

Peng W (1997) Automated generalization in GIS. Ph.D. thesis, Wageningen Agricultural University and International Institute for Aerospace Survey and Earth Science (ITC), Enschede, Hollande

Rigaux P, Scholl M (1995) Multi-scale partitions: Applications to spatial and statistical databases. In: Proceedings of the 4th International Symposium on Advances in Spatial Databases, SSD'95, LNCS 951. Springer-Verlag, pp 170-183

Stell S, Worboys M (1998) Stratified Map Spaces: A formal basis for multi-resolution spatial databases. In: Proceedings of the 8th International Symposium on Spatial Data Handling. pp 180-189

Timpf S (1998) Hierarchical structures in map series. Ph.D. thesis, Technical University Vienna

Vangenot C (2001) La multi-représentation dans les bases de données géographiques (in French). Ph.D. thesis, no.2430, EPFL

Weibel R, Dutton G (1999) Generalizing spatial data and dealing with multiple representations. In: Longley P, Goodchild MF, Maguire DJ, Rhind DW, Geographical Information Systems: Principles, Techniques, Management and Applications, (1), 2nd Edition. Geoinformation International