Multi-focus Research and Geospatial Data - anthropocentric concerns

André Santanchè[1], João Sávio C. Longo[1], Geneviève Jomier[2], Michel Zam[2], Claudia Bauzer Medeiros[1]

[1] University of Campinas, Brazil
{santanche, cmbm}@ic.unicamp.br, joaosavio@lis.ic.unicamp.br
[2] University Paris-Dauphine
{genevieve.jomier, zam}@dauphine.fr

Abstract. Work on multiscale issues presents countless challenges that have been long attacked by GIScience researchers. Research is usually concentrated in one of two directions - new data models to support handling multiple scales, or data structures and algorithms to process data across scales. Complementary implementation aspects are concerned with generalization (and/or virtualization of distinct scales), or with linking entities of interest across scales (e.g., using bottom-up implementation of specific structures, without relying on any specific DBMS). However, researchers seldom take into account the fact that multiscale scenarios are increasingly constructed cooperatively, and require distinct perspectives of the world, in which each research group considers specific aspects of a problem. The combination of handling multiple scales at a time, and having multiple user perspectives per scale constitutes what we call *multi-focus* research. This paper presents our proposal to attack multi-focus scenarios, which considers distinct aspects of the problem of managing multiple scales, illustrated with examples of multiscale geospatial data. Our approach builds upon a specific database version model – the so-called multiversion MVDB – which has already been successfully implemented in several geospatial scenarios, being extended here to support multi-focus research. This extension was implemented and tested in a real world case study, briefly discussed here.

Categories and Subject Descriptors: H.2 [**Database Management**]: Miscellaneous; H.3 [**Information Storage and Retrieval**]: Miscellaneous

Keywords: geospatial data, multiple aspects, multiscale, version

1. INTRODUCTION

Issues involving multiscale data management can be found in most scientific domains, and are associated with data-intensive science. For instance, in February 2011, Science[1] dedicated an entire issue to challenges of handling scientific data and the data deluge. There, 23 articles from experts working in many scientific fields exemplify open problems of dealing with data management, indexing, analysis and visualization. In all of these papers, one can find problems concerning analysis of multiple, interacting space-time scales. Regardless of the theme, there are a few common concerns. The first one is data availability and sharing – and thus problems of consistency, security, privacy and curation. The second is the need for new data analysis and visualization mechanisms, with emphasis on data evolution through time. A third concern involves handling heterogeneity – of data and of expert domains – and hence, the issue of multiple interacting abstraction levels.

[1] Dealing with Data – Challenges and Opportunities – Special Issue. *Science*, vol. 331, Feb 2011, http://www.sciencemag.org/site/special/data/

This work was partially supported by the AMIB binational research project (CAPES-COFECUB), as well as by grants from Brazilian agencies FAPESP and CNPq, in particular the FAPESP-MSR Virtual Institute (NavScales project) and the CNPq MuZOO project, as well as FAPESP/Cepid in Computational Engineering and Sciences, FAPESP-PRONEX (eScience) and the INCT in Web Science.

A large percentage of these papers is related to the concept of "anthropocene-linked research". The emphasis of such research is to study the impact of humans on the environment, and vice versa. Geological societies, all over the world, are adopting the term "Anthropocene" to designate a new geological epoch whose start coincides with the impact of human activities on the Earth's ecosystems and their dynamics. The name was coined in 2000 by Nobel prize chemist Paul Crutzen, who proposed that the Anthropocene starts with the invention of steam engines. Others claim that this has to be pushed back to the first human settlements, thousands of years ago. Regardless of when it started, related studies rely heavily on the analysis of multiscale spatio-temporal data. Moreover, there are multiple perspectives under which a given region can be analyzed, for a specific geographic scale and time period.

For such scenarios, one can no longer consider data heterogeneity alone, but also the heterogeneity of processes that occur within and across scales. Such heterogeneity is intrinsic to the need to combine many kinds of expertise to solve a problem. This is complicated by the following: (a) there are distinct fields of knowledge involved (hence different data collection methodologies, models and practices); and (b) the study of complex systems requires complementary ways of analyzing a problem, looking at evidence at distinct aggregation/generalization levels – a *multi-focus* approach. Since it is impossible to work at all foci at once, each group of scientists will concentrate on a given (sub)problem and try to understand its complex processes. The set of analyses performed under a given focus has implications on others. From now on, this paper will use the term "multi-focus" to refer to these problems, where a "focus" is a perspective of a problem, including data (and data representations), but also modeling, analysis and dynamics of the spatio-temporal entities of interest, within and across scales. It must be stressed that multi-focus research *is not restricted* to geospatial anthropocene phenomena; this paper, however, will concentrate on such phenomena.

Let us now provide a few examples of the challenges in multi-focus anthropocene research. A good example is transportation planning. At a given granularity, engineers are interested in individual vehicles, for which data are collected (e.g., itineraries). Other experts may have a distinct focus for the same data – they store and query trajectories, and associate semantics to stops. At a higher level, traffic planners study trends - the individual vehicles disappear and the entities of study become clusters of vehicles and/or traffic flow [Medeiros et al. 2010]. A complementary focus comes from climate research (e.g., floods cause major traffic disturbances). Cultural habits (and thus social science studies) can also interfere with traffic and produce particular traffic patterns across a country. This can be generalized to several interacting granularity levels. In spite of advances in transportation research involving spatial data, e.g., in moving objects, there are very few results in representation and interaction of multiple foci.

Global warming presents a different set of challenges to multi-focus work. Studies consider a hierarchy of ecological levels, from community to ecosystem, to landscape, to a whole biome. Though ecosystems are often considered closed systems for study purposes, the same does not apply to landscapes, e.g., they can include rivers that run into (or out of) boundaries (similar to studies in traffic in and out of a region...). A landscape contains multiple habitats, vegetation types, land uses, which are inter-related by many spatio-temporal relationships. A given study may target (i.e., focus) vegetation patches, while another will concentrate on the impact of cattle raising in desertification.

In agriculture – the running example in this paper – the focus varies from sensor to satellite data, analyzed under land use practices. Another focus would be to use the same data to study the response of crop strains to climate variables, and still another to economic implications of labor practices in a region – while the former is related to research in agriculture, the latter involves social scientists, for widely distinct analyses and models. Each of the disciplines involved has its own work practices, which require analyzing data at several granularity levels; when all disciplines and data sets are put together, one is faced with a highly heterogeneous set of data and processes that vary on space and time, and for which there are no consensual storage, indexation, analysis or visualization procedures.

The previous examples illustrate our concept of focus – it corresponds to a perspective of study of a given problem, where data can be restricted to one specific scale/representation, or put together objects from distinct scales. Moreover, given the same set of data, distinct foci will arise when the data are analyzed under different models, processed using focus-specific algorithms, or even visualized with particular means. This scenario opens a wide range of new problems to be investigated [Longo et al. 2012; Longo and Medeiros 2013]. This paper has chosen to concentrate on the following challenges, all concerning anthropocene problems:

—How can research on spatial data provide support to research that is characterized by the need to analyze data, models, processes and events at distinct space and time scales, and represented at varying levels of detail, and studied under multiple foci?

—How to keep track of events as they percolate bottom-up, top-down and across space, time and foci of interest?

—How to provide adequate management of these multi-focus multi-expertise scenarios and their evolution?

Previous work of ours in traffic management, agriculture and biodiversity [Medeiros et al. 2010; Longo and Medeiros 2013] brought to light the limitations of present research on spatio-temporal information management, when it comes to supporting multi-focus studies. As will be seen, our work combines the main solution trends found in the literature, handling both data and processes in a homogeneous way, expanding the paradigm of *multiversion databases*, under the MVDB model of Cellary and Jomier [1990]. The multiversion database (MVDB) model has been already implemented to support several geospatial applications [Peerbocus et al. 2004]. This paper extends the work published by Santanche et al. [2012] where we lay the grounds for our proposal to support multi-focus research by extending the multiversion database paradigm. Though this paper is directed to anthropocene-related geospatial data issues, many of its central ideas – in particular, concerning multi-focus research – can be generalized to other multi-focus fields.

2. RELATED WORK

Research on multi-focus data management involves state-of-the-art work in countless fields. For instance, protein structure prediction and folding is a canonical example in which computer scientists, biologists, and theoretical physicists and chemists have joined efforts. Another example is found in the recent development of multiscale simulation approaches, which draws on finite element methods of computational engineering and atomistic simulation techniques, well-known to computational physicists and chemists, with direct impact in, for instance, nanotechnology and to the study of materials and systems across multiple space and time scales. Chemoinformatics is yet another recent and fast growing field in which computer scientists and chemists combine their expertise to develop a variety of methods and tools designed for applications in the areas of drug design and modeling of complex chemical reactions. Multidisciplinary research involving multiple foci is also at the heart of multinational initiatives, such as the MAPPER project [MAPPER 2007], where teams of scientists from several countries are beginning to investigate multiscale and multi-focus issues in five areas: fusion, clinical decision making, systems biology, nano science, and engineering.

Since a "focus", in this paper, is defined as a perspective of study that requires selecting and tailoring the data of interest to a group of users/application, there are many fields in Computer Science where this concept is considered, in particular in databases, complex networks (and graph databases) and software engineering (e.g., use-case diagrams can be seen as a means to specify a focus).

In databases, a perspective that is constructed on top of a database is often treated under the *view* paradigm. Though a view was originally defined to be the result of a query, its definition has evolved with time to designate a portion of the data that is of interest to a specific group of users – and thus, close to our focus definition. Research on views started in the 70's and ranges from view specification and construction-materialization to mapping updates. Views are also involved in query optimization strategies, and in security concerns [Furtado et al. 1979; Stonebraker et al. 1990; Medeiros et al. 2000; Halevy 2001; Olivier et al. 2008]. This original notion is now being extended to new database models, in particular views on graph databases [Fan et al. 2014]. Interactions across views are also studied under many guises, for distinct kinds of database models.

Research in multiple foci is also discussed under the generic umbrella of "complex networks", often analyzed at multiple aggregation levels. Complex networks are used as a unifying framework to study a wide range of domains, from social networks [King 2011] to biodiversity, e.g., the so-called trophic networks. Not only do these networks evolve with time, as relationships (edges across nodes) change, but node aggregations also appear and disappear dynamically. In the context of this paper, a sub-network, or even a node aggregation, might be considered a focus, created to meet the requirements of a specific user or set of users, given the underlying base data. Many researchers are investigating the implementation of complex networks using graph databases. Here, a particular challenge is finding out new constructs, in such databases, to allow creating and joining arbitrary sub-graphs to meet a focus constraint [Daltio and Medeiros 2014].

This paper concentrates on geospatial data issues, where spatial and temporal scales are examples of particular foci, but where the multi-focus perspective occurs in any such scale. As pointed out by Spaccapietra et al. [2002], multiple cartographic representations are just one example of the need for managing multiple scales. In climate change studies, or agriculture, for instance, a considerable amount of the data are geospatial – e.g., human factors.

While research on foci, in the geospatial sense, is usually restricted to representation issues, there is a vast amount of literature when focus is synonym to geographical scale. Present research on multiscale issues has several limitations in this broader scenario. To start with, in the geospatial context, it is most frequently limited to vectorial data, whereas many domains, including agriculture, require other kinds of representation and modeling (including raster data) [Leibovicia and Jackson 2011]. Also, it is essentially concerned with the representation of geographic entities (in special at the cartographic level), while other kinds of requirements must also be considered.

The example reported by Benda et al. [2002], concerning riverine ecosystems, is representative of challenges to be faced and which are not solved by research on spatio-temporal data management. It shows that such ecosystems involve, among others, analysis of spatio-temporal data and processes on human activities (e.g., urbanization, agricultural practices), on hydrologic properties (e.g., precipitation, flow routing), and on the environment (e.g., vegetation and aquatic fauna). This, in turn, requires cooperation of (at least) hydrologists, geomorphologists, social scientists and ecologists. An event at a given focus (e.g., change of agricultural practices) will impact others (e.g., sediment transport and deposition, impacting erosion, fauna and flora at a much larger scale).

Multi-scale analyses are also performed to cope with large amounts of data, by aggregating data in space and/or time. For instance, the work of García et al. [2013] aggregates trajectory segments to be able to compute positional properties of moving objects. Here, new scales are artificially created to help derive useful information (e.g., rather than looking at individual points, regions of interest are created by aggregating sets of points). Still another facet of the same kind of issue concerns a combination of focus and multi-resolution queries. For instance, the work of Nutanong et al. [2012] proposes query and visualization mechanisms to process geospatial data, so that users can zoom into a given region, and select objects of interest (i.e., focus) when that region contains hundreds (or even millions) of overlapping objects. Here, zooming in and out use several generalization algorithms, whereas selection is based on a combination of filtering and sampling.

There are three basic approaches to managing multi-scale spatial data: (a) store data at just one scale, and compute other scales on the fly, e.g., using generalization; (b) store detailed data at each scale of interest, and maintain each separately; (c) a hybrid approach, in which a few scales are stored and others are computed. As will be seen, our approach to this spatial scale issue follows the third phylosophy, and takes advantage of a database versioning model to manage multi-scale data. Generalization algorithms are mostly geared towards handling multiple spatial scales via algorithmic processes; multi-representation databases (MRDBs) are geared towards approach (b), supporting data management at multiple spatial scales. Generalization and MRDBs respectively correspond to Zhou and Jones [2003] multi-representation spatial databases and linked multi-version databases[2]. Most solutions, nevertheless, concentrate on spatial "snapshots" at the same time, and frequently do not consider evolution with time or variation of focus.

Generalization-based solutions rely on the construction of virtual spatial scales from a basic initial geographic scale, Oosterom and Stoter [2010] in their model mention that managing scales require "zooming in and out", operations usually associated with visualization (but not data management). Here, as pointed out by Zhou and Jones [2003], scale and spatial resolution are usually treated as one single concept. Generalization itself is far from being a solved subject. As stressed by Buttenfield et al. [2010], for instance, effective multiscale representation requires that the algorithm to be applied be tuned to a given region, e.g., due to landscape differences. Generalization solutions are more flexible than MRDBs, but require more computing time.

While generalization approaches compute multiple virtual scales, approaches based on data structures rely on managing stored data. Options may vary from maintaining separate databases (one for each scale) to using MRDBs.The latter concern data structures to store and link different objects of several representation of the same entity or phenomenon [Sarjakoski 2007]. They have been successfully reported in, for instance, urban planning, or in the aggregation of large amounts of geospatial data and in cases that applications require data in different levels of detail [Oosterom 2009; Gao et al. 2010; Parent et al. 2009]. The multiple representation work of Oosterom and Stoter [2010] comments on the possibility of storing the most detailed data and computing other scales via generalization. This presents the advantage of preserving consistency across scales (since all except for a basis are computed), but multiple foci cannot be considered.

Our motivation for adopting the hybrid approach is twofold. First, it supports computing new objects at a given scale, thereby supporting arbitrary foci. Second, multiscale data are materialized in a few choice scales due to modelling requirements and application efficiency, which influence the best scale to be used [Bertino et al. 2010]. Depending on the case, it may be necessary to vary the scales for better data visualization or for different types of analysis, which can result in loss of valuable information. It is up to the user to define which scales to materialize, and which to derive. For instance, consider two different spatial scales A and B such that A is larger than B. In cartography, this means that ach object B will contain one or more corresponding objects of A, but the reverse may be not true [Camossi et al. 2008]. Moreover, inconsistencies may occur by varying the scale. Modifications in objects in a scale (e.g., geometry, localization) can make the data in other scales inconsistent. The same remarks apply to scaling in time.

The previous paragraphs discussed work that concentrates on spatial, and sometimes spatio-temporal issues[3]. Several authors have considered multiscale issues from a conceptual formalization point of view, thus being able to come closer to our focus concept. An example is the work of Spaccapietra et al. [2002], which considers classification and inheritance as useful conceptual constructs to conceive and manage multiple scales, including multiple foci. The work of Duce and Janowicz [2010] is concerned with multiple (hierarchical) conceptualizations of the world, restricted to spatial administrative boundaries (e.g., the concept of rivers in Spain or in Germany). While this is related to our problem (as multi-focus studies also require multiple ontologies), it is restricted to ontology construction. We, on the other hand, though also concerned with multiple conceptualizations of geographic space, need

[2] We point out that our definition of *version* is not the same as that of Zhou and Jones

[3] The notion of scale, more often than not, is associated with spatial resolution, and time plays a secondary role.

to support many views at several scales – e.g., a given entity, for the same administrative boundary, may play distinct roles, and be present or not.

We point out that the work of Parent et al. [2006] concerning the MADS model, though centered on conceptual issues concerning space, time and perspective (which has similar points with our focus concept), also covers implementation issues in a spatio-temporal database. Several implementation initiatives are reported. However, a perspective (focus) does not encompass several scales, and the authors do not concern themselves with performance issues. Our extension to the multiversion database (MVBD) approach, discussed next, covers all these points, and allows managing both materialized and virtual data objects within a single framework, encompassing both vector and raster data, and letting a focus cover multiple spatial or temporal scales.

3. RUNNING EXAMPLE

Let us briefly introduce our running example – agricultural monitoring. In this domain, phenomena within a given region must be accompanied through time. Data to be monitored include, for instance, temperature, rainfall, but also soil management practices, and even crop responses to such practices. More complex scenarios combine these factors with economic, transportation, or cultural factors.

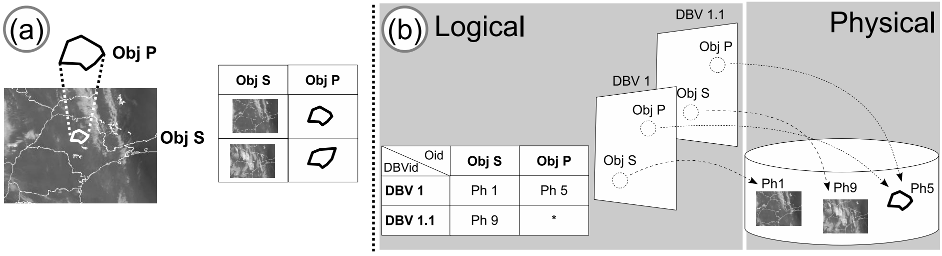
Data need to be gathered at several spatial and temporal scales – e.g., from chemical analysis on a farm's crop every year, to sensor data every 10 minutes. Analyses are conducted by distinct groups of experts, with multiple foci – agro-environmentalists will look for impact on the environment, others will think of optimizing yield, and so on.

We restrict ourselves to two data sources that vary with time, satellite images (typically, one image every 10 days) and ground sensors, abstracting details on the actual data being produced. These sources are analyzed against a background of non-evolving data, namely, vectorial data on the region itself – e.g, topography, waterways. From a high level perspective, both kinds of dynamic sources give origin to *time series*, since they periodically produce data that are stored together with timestamps. We point out that these series are very heterogeneous. Sensor (stream) series data are being studied under distinct research perspectives, in particular data fusion and summarization [McGuire et al. 2011]. Some of these methods are specific for comparing entire time series, while others can work with subsequences. Satellite images are seldom considered under a time series perspective: data are collected less frequently, values are not atomic, and processing algorithms are totally different – research on satellite image analysis is conducted within remote sensing literature [Xavier et al. 2006]. Our multi-focus approach, however, can treat both kinds of data sources homogeneously.

Satellite time series are usually adopted to provide long-term monitoring, and to predict yield; sensor time series are reserved for real time monitoring. However, data from both sources must be combined to provide adequate monitoring. Such combinations present many open problems. The standard, practical, solution is to aggregate sensor data temporally (usually producing averages over a period of time), and then aggregate them spatially. In the spatial aggregation, a local sensor network becomes a point, whose value is the average of the temporal averages of each sensor in the network. Next, Voronoi polygons are constructed, in which the "content" of a polygon is this global average value. Finally, these polygons can be combined with the contents of the images. Joint time series evolution is not considered. Our solution, as will be seen, allows to solve these issues within the database itself – rather than creating external software layers to process data.

4. SOLVING ANTHROPOCENIC ISSUES USING MVDBS

Our solution is based on the Multiversion Database (MVDB) model, which will be only introduced in an informal way. For more details the reader is referred to the work of Cellary and Jomier [1990]. The solution is illustrated by considering the monitoring of a farm within a given region, for which



Oid / DBVid	Obj S	Obj P
DBV 1	Ph 1	Ph 5
DBV 1.1	Ph 9	*

Fig. 1. (a) Practical scenario of a polygon over a satellite image; (b) The relationship between DBVs, logical and physical identifiers.

time-evolving data are: (a) satellite images (database object S); (b) the farm's boundaries (database object P), and (c) weather stations at several places in the region, with several sensors each (database object G).

4.1 Introducing MVBD

Intuitively, a given real world entity can correspond to many distinct digital items expressing, for example, its alternative representations, or capturing its different states along time. Each of these "expressions" will be treated in this work as a *version* of the object. Consider the example illustrated in Figure 1(a). On the left, there are two identified database objects: a satellite image (`Obj S`) and a polygon to be superimposed on the image (`Obj P`). delimiting the boundaries of the farm to be monitored.

As illustrated by the table on the right of the figure, both objects can change along time, reflecting changes in the world, e.g., a new satellite image will be periodically provided, or the boundaries of the farm can change. For each real world entity, instead of considering that these are new database objects, such changes can be interpreted as many versions of the same object[4]. This object has a single, unique, identifier – called an Object Identifier `Oid`[5].

A challenge when many interrelated objects have multiple versions is how to group them coherently. For example, since the satellite image and the farm polygon change along time, a given version of the satellite image from 12/05/2010 must be related with a temporally compatible version of the farm polygon. This is the central focus of the Multiversion Database (MVDB) model. It can handle multiple versions of an arbitrary number of objects, which are organized in *database versions - DBVs*. A DBV is a logical construct. It represents an entire, consistent database constructed from a MVDB which gathers together consistent versions of interrelated objects. Intuitively, it can be interpreted as a *complex view* on a MVDB. However, as shall be seen, unlike standard database views, DBVs are not constructed from queries.

To handle the relation between an object and its versions, the MDBV model distinguishes their identifications by using object and physical identifiers respectively. Each object has a single object identifier (`Oid`), which will be the same independently of its multiple versions. Each version of this object, materialized in the database by a digital item – e.g., an image, a polygon etc. – will receive a distinct physical version identifier *PVid*. In the example of Figure 1(a), there is a single `Oid` for each object – satellite image (`Obj S`) and the farm boundaries (`Obj P`). Every time a new image or a new polygon is stored, it will receive its own `PVid`.

DBVs are the means to manage the relationship between an `Oid` (say, `S`) and a given `PVid` (of `S`). Figure 1(b) introduces a graphical illustration of the relationship among these three elements: `DBV`,

[4] Here, both raster and vector representations are supported. An MVDB object is a database entity

[5] Oids are artificial constructs. The actual disambiguation of an object in the world is not an issue here

Oid and PVid. In the middle there are two DBVs identified by DBVids – DBV 1 and DBV 1.1 – and represented as planes containing logical slices (the "views") of the MVDB. The figure shows that each DBV has versions of P and S, but each DBV is monoversion (i.e., it cannot contain two different versions of an object). The right part of the figure shows the physical storage, in which there are two physical versions of S (identified by Ph1 and Ph9), and just one version of P.

DBV 1 relates S with a specific satellite image and P with a specific polygon, which form together a consistent version of the world. Notice that here nothing is being said about temporal or spatial scales. For instance, the two satellite images can correspond to images obtained by different sensors aboard the same satellite (e.g., heat sensor, water sensor), and thus have the same timestamp. Alternatively, they can be images taken in different days. The role of the DBV is to gather together compatible versions of its objects, under whichever perspective applies.

Since DBVs are logical constructs, each object in a DBV has its own logical identifier. Figure 1(b) shows on the left an alternative tabular representation, in which DBVids identify rows and Oids identify columns. Each pair (DBVid, Oid) identifies the logical version of an object and is related to a single PVid, e.g., $(DBV1, ObjS) {\rightarrow} Ph1$. The asterisk in cell (DBV 1.1, Obj P) means that the state of the object did not change from DBV 1 to DBV 1.1, and therefore it will address the same physical identifier Ph 5.

4.2 DBV Evolution and Traceability

DBVs can be constructed from scratch or from other DBVs[6]. The identifier of a DBV (DBVid) indicates its derivation history. This is aligned to the idea that versions are not necessarily related to time changes, affording alternative variations of the same source, as well as multiple foci – see section 5.

The distinction between logical and physical identifications is explored by an MVDB to provide storage efficiency. In most of the derivations, only a partial set of objects will change in a new derived DBV. In this case, the MVDB has a strategy in which it stores only the differences from the previous version. Returning to the example presented in Figure 1(b) on the left table, DBV 1.1 is derived from DBV 1, by changing the state of Obj S. Thus, a new PVid is stored for it, but the state of Obj P has not changed – no new polygon is stored, and thus there is no new PVid.

The evolution of a DBV is recorded in a derivation tree of DBVids. To retrieve the proper PVid for each (virtual) object in a DBV, the MVDB adopts two strategies: provided and inferred references[7], through navigation in the tree. This allows keeping track of real world evolution. We take advantage of these concepts in our extension of the MVDB model, implemented to support multiple spatial scales [Longo et al. 2012]. First, we create one tree per spatial scale, and all trees grow and shrink together. Second, the notion of object id is extended to associate the id with the scale in which that object exists - (Oid, Scaleid). This paper extends this proposal in two directions: (1) we generalize the notion of spatial scale to that of focus, where a given spatial or temporal scale can accomodate multiple foci, and the evolution of these foci within a single derivation tree; (2) we provide a detailed case study to illustrate the internals of our solution.

5. FROM MULTIVERSION TO MULTI-FOCUS

5.1 From Multiversion to Multiscale

The extension of the MVBD model to support multi-focus is presented in two steps. First, we introduced a modification in this model to support multiple scales in space and time [Longo 2013]. Here,

[6] DBV derivation trees, part of the model, will detailed in section 6.

[7] For the logical version (DBV 1.1, Obj P), the reference will be inferred by traversing the chain of derivations.

the main modifications were the following: (a)for each scale, a new derivation tree is constructed, and changes in the world in a given scale are synchronized across version trees; and (b) a scale becomes part of an oid, The first modification implies that a given DBV (containing a set of objects) can be created in several scales (where all objects are consistent with that scale). The second modification indicates that the logical version of an object requires not only the identification of its DBV, but also of the scale under which it is being studied.

This work was implemented and tested on vectorial data – see section 6. An additional characteristic of this implementation is that integrity constraints can be specified and checked between two consecutive (stored) scales. In this extension to the MVBD model, the notion of focus is restricted to a scale – i.e., it does not support a focus in which people work in subsets of objects, or with objects constructed dynamically. The only relationships between objects (or scales) are spatial (topologic, direction) or temporal. We now describe the next step, in which this is extended to the full multi-focus approach.

5.2 From Multiscale to Multi-focus

This paper extends the MVDB model to support the several flavors of multi-focus. This implies in synthesizing the multiple foci which can be applied to objects – scales, representations etc. – as specializations of versions. Figure 2 illustrates an example of this extension. There are three perspectives within the logical view - see the Figure.

In the Physical perspective, there are three objects – two versions of satellite image S (with identifiers `Ph1` and `Ph2`), and one version of a set of sensor data streams, corresponding to a set of weather stations G – global identifier `Ph7`). Satellite image and sensor data are to be combined by Applications, which can only access them through DBVs – i.e. they cannot address them straight in the Physical Storage. So, several DBVs are built, each of which corresponding to a distinct focus. The arrows between DBV objects and stored objects appear whenever a logical object has a provided reference (not inferred or computed) to a physical object – it means that this object was updated in this DBV. In the figure, the DBV corresponding to Focus 1 makes available the satellite image version `Ph1` and all data from all weather stations G. The DBV corresponding to Focus 2 makes available the satellite image version `Ph2`, and *computes* a set of Voronoi polygons from the weather station data streams – the resulting polygon is displayed in the figure with a dotted line to show that it is not directly copied from the database, but is computed from it. Finally, DBV-Focus3 contains only one image, which has been computed from DBV-Focus2.

Applications access these three DBVs in the following way. Application Scale A is built from DBV-Focus1; it corresponds to a particular spatio-temporal focus of the database, in which the image and the polygons are directly copied from the DBV. Application Scale B is built from DBV-Focus2; it corresponds to another spatio-temporal focus of the database, in which the image is directly extracted from the DBV and a set of Voronoi polygons is computed from the DBV. The third DBV is not being used by any application. Additional data sources of our running example are added the same way – e.g., our static data sources can be added for each scale (A or B).

Figure 2 reflects the following facts. First, DBVs can contain just objects that are in the database, or computed objects, or a mix of both. Second, applications constructed on top of the DBVs can use exactly the same objects (object S on Scale A / DBV-Focus1 directly uses the same contents of DBV-Focus2), but also compute other objects (the Voronoi polygons on Scale B / DBV-Focus2, computed from DBV-Focus1). Third, DBVs now can be interrelated through many kinds of derivation operations.

In our case study, each application corresponds to one spatial scale (scale B smaller than scale A), and sensor data are preprocessed either at the application, or by the DBMS, to allow combination of these distinct data sources. DBV-Focus 3 is an example of at least three possible scenarios: in

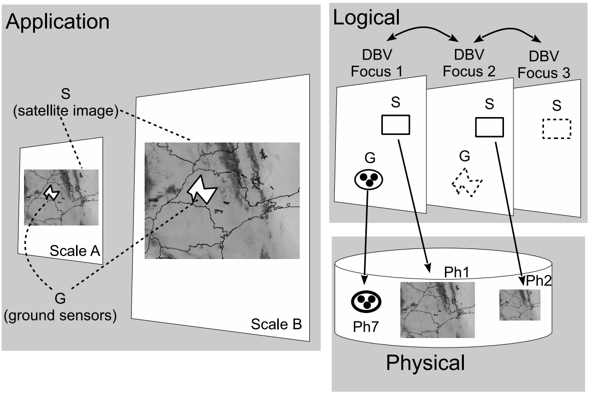


Fig. 2. Handling multiple foci

one, S corresponds to an even smaller spatial scale, for which sensor data do no longer make sense; in another, S becomes the result of combination of satellite image and sensor data; in the third, the focus concentrates in some characteristics of the satellite image, and sensor data can be ignored for the purposes of that DBV.

In order to support multiple foci, our multi-scale extension of the MVDB model was extended in two aspects: (i) we added more types of relationships between DBVs; (ii) we introduced an extended strategy to infer and derive values, and compute logical object versions. In the classical MVDB the only relationship between two DBVs is the derivation relationship. Our multi-focus approach requires a wider set of relationships. Therefore, now the relationship between two DBVs is typed. Besides derivation, new types are created: generalization, aggregation etc. This typing system is extensible, affording new types. This requires that new information be stored concerning each DBV, and that the semantics of each object be stored alongside the object, e.g., using ontologies.

Different from the classical inference mechanism adopted for derivations, these new relationship types require new inference algorithms. For example, a digital item from a scale A must be resized to a scale B. The inference will be influenced by four variables: (1) the types of the involved DBVs; (2) the type of the relationship between them; (3) the datatype of the digital item; (4) the semantics in which the object will be handled. This requires that new information be stored concerning each DBV, and that the semantics of each object be stored alongside the object, e.g., using ontologies.

Returning to our example in Figure 2 consider an application that will access the contents of `S` in `DBV-Focus3`. Since there is no explicit reference to it in the `DBV-Focus2`, the only information is that the state of S in the third focus has been derived in some kind of relationship with the state of S in the second DBV. Let us consider that this is a generalization relationship, i.e., the state of S in the third DBV is a cartographic generalization of the state of S in the DBV-Focus2. In order to use this logical version of S in an application, the construction of DBV-Focus3 will require an algorithm that will: (1) verify that the type of the relationship between DBVs is generalization and that they are typed by geographic scales; therefore, `S` must be transformed to the proper scale; (2) check the semantics of `S` and its datatype, verifying that it is a raster satellite image, and therefore generalization concerns image processing, and scaling.

Figure 3 shows the UML diagram of the data model that supports our multi-focus paradigm; this diagram was specialized to support management of inter-scale integrity constraints [Longo 2013]. This specialization is a first approach to evaluate some of the proposed features, with some restrictions: (i) the DBV typing system is still restricted to spatial scales; (ii) the object typing system is still restricted to geo-objects but we provide an extension mechanism.

Physical versions of objects can be defined as subclasses of the *GenericPV* class and will not

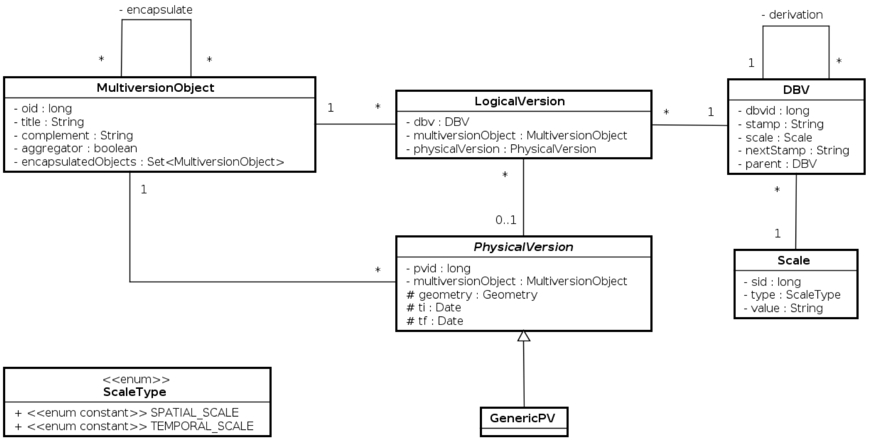


Fig. 3. UML diagram of the multi-focus model

be discussed here; this helps support multiple kinds of relationship across physical versions. The *MultiversionObject* class has five attributes. The first is the identifier, the second is some title which identifies the object in the real world, the third is some complement of the *title* attribute, the fourth says if the object is an aggregator, and the fifth is the set of encapsulated objects if this is an aggregator.

The *Scale* class has an identifier, an attribute that indicates the type of the scale (spatial, temporal, etc) and another that is the value associated to the type (e.g., "1:10000" for spatial scales, "minutes" for temporal scales).

The *DBV* class has five attributes. The first is the identifier, the second is the version *stamp*, the third is the associated scale, the fourth stores the *stamp* of the next child to be created by derivation (e.g., if a DBV with stamp 0.1 has already two children – 0.1.1 and 0.1.2 – the next child attribute will indicate 0.1.3), and the fifth is the DBV from which it was derived. The *PhysicalVersion* abstract class has an identifier, a geometry, and an initial and final timestamp: t_i and t_f, respectively. The *GenericPV* subclass has no aditional data besides *PhysicalVersion* attributes (users can create other subclasses of *PhysicalVersion* entering new attributes to be versioned). Finally, the *LogicalVersion* class links a DBV and a multiversion object with a *physical version*.

In order to support multiple kinds of relationships across DBVs, as described, the *d*erivation (self)relationship among DBVs in the Figure is extended to accommodate all intended derivation types. Moreover, class *S*cale represents all kinds of focus.

6. IMPLEMENTATION OF MULTISCALE MANAGEMENT

We implemented a multiscale data management platform, for geospatial data, using our extension of the DBV model to handle multiple scales – see section 5.2 This implementation does not consider multiple relationships across DBVs (which is our modeling approach to full multi-focus support); it only treats multi-scale modeling and storage. In other words, this implementation only gives limited support to specific kinds of focus, namely when a focus corresponds to objects within a single spatial or temporal scale.

This work, detailed by Longo [2013], also allows the specification and checking of multi-scale integrity constraints, specified on spatial (vectorial) and temporal scales. Here, we give a brief overview of this implementation, showing how our proposal can be implemented using a database versioning mechanism. In this implementation, a focus is created from all objects within a single scale, selected via their oids (that include the scale identifier). It does not allow to restrict the set of objects to be

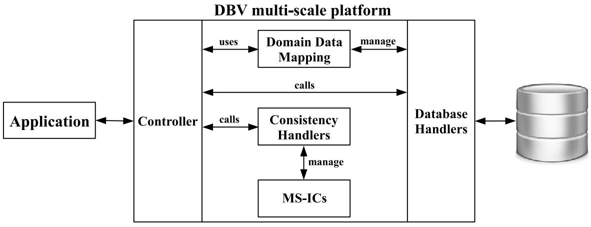


Fig. 4. Architecture of the DBV multi-scale platform

managed (though this can be changed by constructing views on top of DBVs). The implementation does not support the computation of virtual objects, or arbitrary relationships across DBVs. It supports, however, scaling relationships specified as multiscale integrity constraints.

The platform[8] was implemented on top of the PostGIS spatial database extension for PostgreSQL due to its widespread adoption and to its support of geospatial features. Our implementation uses the Java programming language, Java Persistence API (JPA) and Generic Spatial DAO library, which is a generic DAO (Data Access Object) with spatial extensions (using Hibernate Spatial) and utility methods, for geographic data object/relational mapping.

Figure 4 shows a high level model of the platform, divided in five modules: *Domain Data Mapping*, *Database Handlers*, *Consistency Handlers*, *Multi-Scale Integrity Constraints* and *Controller*.

There follows a brief description of each module:

—*Controller* – accessed by applications to select the DBVs to use and to perform operations on DBVs and objects, as well as to request constraint checking. It plays the role of mediator among the platform's modules, and between the platform and external applications;

—*Domain Data Mapping* – maps application objects into the underlying DBMS;

—*Database Handlers* – translate *Controller* requests into database operations (queries and updates), considering data mappings;

—*MS-ICs* – implements multiscale integrity constraints as Java classes. Each class has a method that checks the multiscale consistency between two versions of an object, i.e., one version per scale, and another that checks the multiscale consistency between two pairs of objects (one pair per scale).

—*Consistency Handlers* – code that checks multi-scale consistency, invoking the appropriate MS-IC methods.

This platform supports, among others, the following: creation of a new DBV, accessing a DBV, adding or modifying a logical version of an object in a DBV, checking multi-scale consistency across DBVs. For implementation details, the reader is referred to the work of Longo [2013].

Visualization and interaction facilities were constructed using the Java programming language, JSF (Java Server Faces) framework, RichFaces visual components for JSF, OL4JSF library, which helps the use of OpenLayers for JSF.

This execution example used multi-scale geospatial real world data provided by Embrapa[9]. The data sets contain 5641 geometry features related to the Rio Pardo watershed and its rivers, at two scales:1:250k and 1:1M. Let us consider the user wants to construct some scenarios using the DBV multi-scale platform, involving watershed polygons at these two scales, and river polylines.

Figure 5 shows the (synchronized) derivation trees of this execution. DBV stamps appear at the top left corner of each DBV. The initial DBVs have no data (by default). The set of DBVs with stamps 0.1

[8] http://code.google.com/p/dbv-ms-platform

[9] http://www.embrapa.br

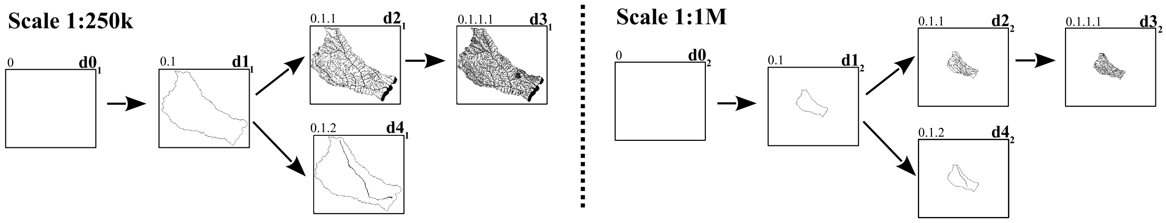


Fig. 5. Derivation trees for the case study

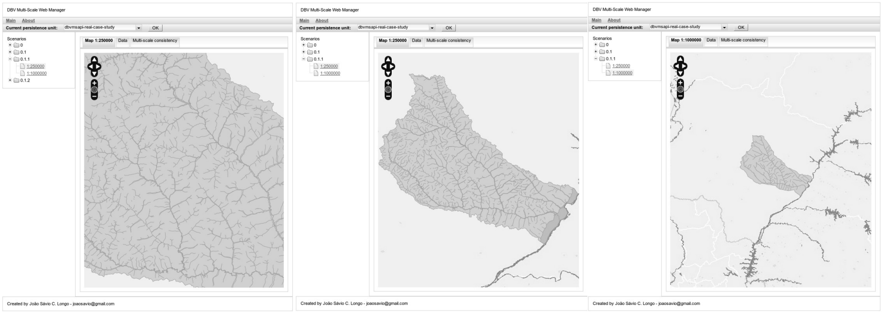


Fig. 6. Scenario corresponding to 0.1.1

has only the watershed polygons in the two scales. For DBVs 0.1.1, we now have the watershed with all rivers in the two scales. Here, we do not have to store the watershed polygons again, because they are shared from the previous DBVs. The same occurs in 0.1.2, where besides the watershed polygons, the main river appears.

The user starts by choosing to create empty DBVs, one for each scale, and then inserts data. Version 0.1 was constructed by creation of a DBV per scale, followed by insertion of objects corresponding to watershed and rivers. This was achieved by execution of operations on multiversion objects and DBVs, via invocation of methods of the platform, e.g., adding or updating versions of objects, and working in a scale at a time. When a new DBV is created from the current, the changes are saved.

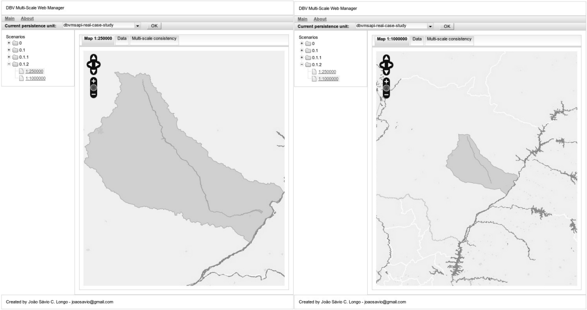
This is achieved by demanding the creation of two new DBVs (one for each scale) descending from DBV(s) stamped 0. The top left side of each screen shows how new versions are progressively created, while the maps portray the actual visualization of multi-scale data at each DBV. At each DBV and scale, the user can visualize more or less details by clicking on the +/- buttons.

Next, suppose the user wants to add data in order to represent two alternative scenarios: one contains the watershed and all its rivers and another with the watershed and only its main river. For each scale, two new DBVs are created descending from versions 0.1. Figures 6 and 7 are screen copies of these two alternative situations, respectively numbered 0.1.1 and 0.1.2.

Figure 6 left shows a zoom of scenario 0.1.1, where river polylines are displayed in more detail.

7. CONCLUSIONS AND ONGOING WORK

This paper presents our approach to handling multi-focus problems, for geospatial data, based on adapting the MDBV (multiversion database) approach to handle not only multiple scales, but multiple foci at each scale. Most approaches in the geospatial field concentrate on the management of multiple spatial or temporal scales (either by computing additional scales via generalization, or keeping track of all scales within a database via link mechanisms). Our solution encompasses both kinds of approach in a single environment, where an *ad hoc* working scenario (the focus) can be built either by getting



Current persistence unit: dbvmsapi-real-case-study

Current persistence unit: dbvmsapi-real-case-study

Created by João Sávio C. Longo - joaosavio@gmail.com

Created by João Sávio C. Longo - joaosavio@gmail.com

Fig. 7. Scenario corresponding to 0.1.2

together consistent spatio-temporal versions of geospatial entities, or by computing the appropriate states, or a combination of both. Since a DBV can be seen as a consistent view of the multiversion database, our approach also supports construction of any kind of arbitrary work scenarios, thereby allowing cooperative work. Moreover, derivation trees allow keeping track of the evolution of objects as they are updated, appear or disappear across scales.

Our ongoing work follows several directions. One of them includes domain ontologies, to support communication among experts and interactions across levels and foci. We are also concerned with continuing our work on formalizing constraints across DBVs (and thus across scales and foci).

We point out that we might also in the future take advantage of another DBV implementation, called MyDraft (www.myDraft.org). Though not constructed for multi-focus purposes, MyDraft is a web platform meant to build and run data-oriented rich web applications. Application designers can define and run executable models (classes, attributes, states, use cases) with little or no code, in incremental, user experience driven, two minutes cycles. Traceability features include automatic history logging and instant time-machine for both data and model definition. Although myDraft is a general purpose platform, geographic applications can be built using chart components. Nevertheless, in order to take advantage of this product, we would have to modify it to support focus manipulation.

REFERENCES

Benda, L. E. et al. How to Avoid Train Wrecks When Using Science in Environmental Problem Solving. *Bioscience* 52 (12): 1127–1136, 2002.

Bertino, E., Camossi, E., and Bertolotto, M. Multi-granular Spatio-temporal Object Models: concepts and research directions. In *Object Databases*. Vol. 5936. Springer Berlin / Heidelberg, pp. 132–148, 2010.

Buttenfield, B., Stanislawski, L., and Brewer, C. Multiscale Representations of Water: tailoring generalization sequences to specific physiographic regimes. In *Proceedings of the Geographic Information Science*. pp. 14–17, 2010.

Camossi, E., Bertolotto, M., and Bertino, E. Multigranular Spatio-Temporal Models: implementation challenges. In *Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*. pp. 63:1–63:4, 2008.

Cellary, W. and Jomier, G. Consistency of Versions in Object-Oriented Databases. In *Proceedings of the International Conference on Very Large Data Bases*. pp. 432–441, 1990.

Daltio, J. and Medeiros, C. B. Handling Multiple Foci in Graph Databases. In *Proceedings of the International Conference on Data Integration in the Life Sciences*. pp. 58–65, 2014.

Duce, S. and Janowicz, K. Microtheories for Spatial Data Infrastructures – Accounting for Diversity of Local Conceptualizations at a Global Level. In *Proceedings of the Geographic Information Science*. pp. 27–41, 2010.

Fan, W., Wang, W., and Wu, Y. Answering Graph Pattern Queries Using Views. In *Proceedings of the IEEE International Conference on Data Engineering*. pp. 167–176, 2014.

Furtado, A., Sevcik, K., and Santos, C. Permitting Updates through Views of Databases. *Informations Systems* vol. 4, pp. 269–283, 1979.

Gao, H., Zhang, H., Hu, D., Tian, R., and Guo, D. Multi-scale Features of Urban Planning Spatial Data. In *Proceedings of the International Conference on Geoinformatics*. pp. 1 –7, 2010.

García, M. G., Ivánová, I., Dilo, A., and Morales, J. Representing Positional Uncertainty of Individual and Aggregated Trajectories of Moving Objects. In *Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*. Orlando, pp. 436–439, 2013.

Halevy, A. Answering Queries using Views: a Survey. *The VLDB Journal* vol. 10, pp. 270–294, 2001.

King, G. Ensuring the Data-Rich Future of the Social Sciences. *Science* 331 (6018): 719–721, 2011.

Leibovicia, D. G. and Jackson, M. Multi-scale integration for spatio-temporal ecoregioning delineation. *International Journal of Image and Data Fusion* 2 (2): 105–119, 2011.

Longo, J. S. C. Management of Integrity Constraints for Multi-scale geospatial Data, 2013. Master Thesis, Supervisor C. B. Medeiros.

Longo, J. S. C., Camargo, L. O., Medeiros, C. B., and Santanche, A. Using the DBV Model to Maintain Versions of Multi-scale Geospatial Data. In *Proceedings of the International Workshop on Semantic and Conceptual Issues in GIS*. pp. 284–293, 2012.

Longo, J. S. C. and Medeiros, C. B. Providing Multi-scale Consistency for Multi-scale Geospatial Data. In *Proceedings of the International Conference on Scientific and Statistical Database Management*. Chicago, pp. 1–12, 2013.

MAPPER. Multiscale Applications on European e-Infrastructures. available at http://www.mapper-project.eu/, 2007.

McGuire, M. P., Janeja, V. P., and Gangopadhyay, A. Characterizing Sensor Datasets with Multi-Granular Spatio-Temporal Intervals. In *Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*. pp. 449–452, 2011.

Medeiros, C. B., Bellosta, M. J., and Jomier, G. Multiversion Views: Constructing Views in a Multiversion Database. *Data & Knowledge Engineering* vol. 33, pp. 277–306, 2000.

Medeiros, C. B., Joliveau, M., Jomier, G., and Vuyst, F. Managing Sensor Traffic Data and Forecasting Unusual Behaviour Propagation. *Geoinformatica* 14 (3): 279–305, 2010.

Nutanong, S., Adelfio, M., and Samet, H. Multiresolution Select-distinct Queries on Large Geographic Point Sets. In *Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*. pp. 159–168, 2012.

Olivier, B., Cohen-Boulakia, S., Davidson, S., and Hara, C. Querying and Managing Provenance through User Views in Scientific Workflows. In *Proceedings of the IEEE International Conference on Data Engineering*. pp. 1072–1081, 2008.

Oosterom, P. Research and Development in Geo-information Generalisation and Multiple Representation. *Computers, Environment and Urban Systems* 33 (5): 303–310, 2009.

Oosterom, P. and Stoter, J. 5D Data Modelling: Full Integration of 2D/3D Space, Time and Scale Dimensions. In *Proceedings of the Geographic Information Science*. pp. 310–324, 2010.

Parent, C., Spaccapietra, S., Vangenot, C., and Zimanyi, E. Multiple Representation Modeling. In *Encyclopedia of Database Systems*, L. Liu and M. T. Ozsu (Eds.). Springer US, pp. 1844–1849, 2009.

Parent, C., Spaccapietra, S., and Zimanyi, E. *Conceptual Modeling for Traditional and Spatio-Temporal Applications - the MADS Approach*. Springer, 2006.

Peerbocus, A., Medeiros, C. B., Voisard, A., and Jomier, G. A System for Change Documentation based on a Spatiotemporal Database. *Geoinformatica* 8 (2): 173–204, 2004.

Santanche, A., Medeiros, C. B., Jomier, J., and Zam, M. Challenges of the Anthropocene Epoch - Supporting Multi-focus Research. In *Brazilian Symposium on Geoinformatics*. pp. 1–10, 2012.

Sarjakoski, L. T. Conceptual Models of Generalisation and Multiple Representation. In *Generalisation of Geographic Information*. Elsevier, pp. 11–35, 2007.

Spaccapietra, S., Parent, C., and Vangenot, C. GIS Databases: From Multiscale to MultiRepresentation. In *Proceedings of the International Symposium on Abstraction, Reformulation, and Approximation*. pp. 57–70, 2002.

Stonebraker, M., Jhingran, A., Goh, J., and Potamianos, S. On Rules, Procedures, Caching and Views in Database Systems. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. pp. 281–290, 1990.

Xavier, A., Rodorff, B., Shimabukuro, Y., Berka, S., and Moreira, M. Multi-temporal Analysis of MODIS Data to Classify Sugarcane Crop. *International Journal of Remote Sensing* 27 (4): 755–768, 2006.

Zhou, S. and Jones, C. B. A Multirepresentation Spatial Data Model. In *Proceedings of the International Symposium in Advances in Spatial and Temporal Databases*. pp. 394–411, 2003.