# Multi-representation Lens for Visual Analytics

Sandro Danilo Gatto[1], André Santanchè[2]

*Instituto de Computação, UNICAMP*
*Caixa Postal 6176 – CEP 13081-970 – Campinas – SP – Brazil*
[1]`sdgatto@ic.unicamp.br`
[2]`santanche@ic.unicamp.br`

***Abstract*— Modern data analysis deeply relies on computational visualization tools, specially when spatial data is involved. Important efforts in governmental and private agencies are looking for patterns and insights buried in dispersive, massive amounts of data (conventional, spatiotemporal, etc.). In Visual Analytics users must be empowered to analyze data from different perspectives, integrating, transforming, aggregating and deriving new representations of conventional as well as spatial data. However, a challenge for visual analysis tools is how to articulate such wide variety of data models and formats, specially when multiple representations of geographic elements are involved. A usual approach is to convert data to a database – e.g., a multi-representation database – which centralizes and homogenizes them. This approach has restrictions when facing the dynamic and distributed model of the Web. In this paper we propose an on the fly and on demand multi-representation data integration and homogenization approach, named Lens, as an alternative that fits better with the Web. It combines a metamodel driven approach to transform data to a unifying multidimensional and multi-representation model, with a middleware-based architecture for seamless and on-the-fly data access, tailored to Visual Analytics.**

## I. Introduction

Visual Analytics (VA) is "the science of analytical reasoning by means of interactive interfaces" [1]. It is multidisciplinary by nature and has risen from the need of turning the overload of information into opportunities to get insights and to make discoveries, even from apparently chaotic data, combining the best what humans have: reasoning and sight. The VA main strategy to cope with human limitation of dealing with large amounts of information involves providing tools that enable analysts to dynamically settle transformations, representations and visualizations.

Interactive and customizable visualization and transformation tools play an essential role in spatiotemporal data analysis, as patterns and information are usually hidden between the lines: there are irregular and convoluted borders; phenomena tend to be specific in scale and are usually correlated in different scales; locations tend to be inaccurate and are scale, resolution and time dependent [1]. Spatiotemporal phenomena may have multiple representations [2], which reflects the different perspectives of users. Analysts can materialize new representations of data, on the fly and on demand, and they can manipulate, publish and share them. Moreover, systems must deal with massive amounts of data, possibly distributed, concerning distinct time, scale, representation and dimensionality, achieving good performance in transformation and retrieval. Therefore, VA is about access, not update, and is about multi-representation.

Works dealing with spatial and multi-representation data (e.g., [3] and [4]) have focused on the database level. The proposed Multi-representation Database (MRDB) models and architectures extends object-relational databases, being centralized and monolithic. Object-relational databases excel at dealing with short, OLTP (online transaction processing), transactions, but are not effective for query-intensive applications, as required by VA. On the other hand, multidimensional databases, usually adopted in OLAP (online analytical processing), can swiftly process analytical queries on huge amounts of data.

VA usually requires data coming from several and distributed sources. They can contain unstructured and heterogeneous data, represented in distinct formats, e.g., shape files, KML and spreadsheets. While the Web is expanding the opportunity to reach data sources, a monolithic central repository to integrate this data may require excessive costs to follow the constant updates and stay consistent and up-to-date. Moreover, in the Web scenario it is not always possible to access an entire data source at once.

These observations motivated our *Multi-representation Lens proposal*. It is an *on the fly* and *on demand* integration approach, in which the original data stay in their origins and are transparently transformed on demand, by a middleware, into the output format apt to querying and analysis. Users specify mappings of data sources to a homogeneous unification model, described in terms of dimensions, facts and measures – a multidimensional data model. It resorts to metamodeling to provide an extensible strategy of model mapping. The "lens" property of our approach facilitates and influences perception, comprehension or evaluation, without changing the original data source itself.

The multidimensional and multi-representation unification model subsidizes building hypercubes (OLAP structures) extended to afford multi-representation, henceforth named multi-representation hypercubes. They can maintain different representations for each data item, including those resulting from transformations users apply. Lens middleware-based architecture is designed to support VA activities. It seamlessly connects any data source – structured and unstructured, local or in the cloud – transparently handling data loading, transformation and keeping the hypercube up-to-date.

This paper presents the design of our Multi-representation

Lens for Visual Analytics architecture and its unifying metamodel, which is the main focus of this paper. It is organized as follows: Section II discusses the requirements expected by a VA framework; Section III discusses the basis of our work; Section IV presents our Lens approach; in Section VI we confront our approach with related work; finally, Section VII concludes this paper and proposes future work.

## II. Architecture Requirements

Lens defines an architecture intended to VA applications. It was designed to meet the following requirements: (1) generic framework, easily adaptable to any domain, (health, security, etc.); (2) flexible to afford a wide variety of analysis and operations; (3) apt to store and manipulate large datasets with good performance; (4) keep source-target database consistency, detecting new versions of data and automatically reflecting such version changes; (5) able to map data – heterogeneous, unstructured and structured (spatial, conventional databases, text, images, sensor data); local or distributed – to a seamless and homogeneous model; (6) able of storing, versioning, sharing and publishing of portrayals defined by users; (7) web-based, enabling different types of clients to access and manipulate data.

As VA comprises several disciplines, the expected functionalities provided by our architecture will articulate and extend those found in original systems, e.g., overlay and spatial queries (GIS); drill-down, roll-up, slice, dice (OLAP); regression, variance, etc. (statistical analysis). Users will be able to plot graphics (pie charts, tag clouds) on top of maps, or re-portray data applying operations of aggregation, transformation and generalization. VA empowers users by combining tools from different disciplines, in order to achieve best results from data. Therefore, users must be able to combine data from different repositories seamlessly. All of these functionalities should be supported by advanced interaction tools, implemented in rich interfaces.

## III. Basic Concepts

This work combines four concepts: the multidimensional/OLAP model; data source metamodeling; data multi-representation; and Visual Analytics. The multidimensional model represents data in terms of dimensions and hierarchies. A multidimensional database is a OLAP database likewise. OLAP is a class of analytical applications that represents data in multiple dimensions, organizing data in "hypercubes" (a cube with possibly more than three dimensions).

In order to afford an expansible approach to align distinct models, Lens describes each involved model and their mappings by using metamodels. The description is represented in OMG MOF (MetaObject Facility) [5] and adopts the OMG CWM Common Warehouse Metamodel [6]. In a nutshell, MOF takes advantage of the UML modeling formalism to specify metamodels (models that specify models). An instance of a metamodel will be a specific model, e.g., there is a metamodel that specifies the relational model; an instance of this metamodel will be a specific relational schema, which is a model. CWM is a set of metamodels specified in MOF aimed to support interchange among different warehouse models (e.g., relational, XML and multidimensional) and transformations among them.

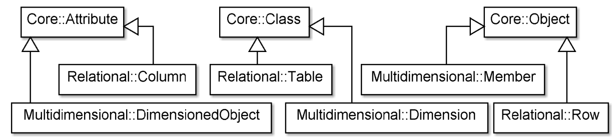


Fig. 1. CWM metamodel of the relational and multidimensional models

Fig. 1 shows a fragment of the CWM metamodel for the relational and multidimensional models. As the figure shows, each class (square) in the metamodel specifies an element of a model. For example, `Table`, `Column` and `Row` are typical elements of the relational model, as `Dimension` and `Member` are for the multidimensional model. The CWM specifies an abstract core metamodel aimed to generalize and bind elements of distinct metamodels. In this sense, `Table` in the relational metamodel specializes the same generic `Class` of `Dimension` in the multidimensional metamodel. This means that they share a role of describing sets of entities following equivalent approaches. It involves, for example, specifying their set of attributes (a relational `Column` and a multidimensional `DimensionObject`). This is the basis for mapping and transformations, which in turn will be modeled by means of CWM Transformation metaclasses (see [7]). Transformation metaclasses can map data structures, e.g., tables to hypercubes, or data processes, e.g., summarization, aggregation and reprojection. Although CWM uses inheritance to relate models, it specifies integrity constraints in OCL (Object Constraint Language), a declarative language for describing rules, that is part of UML specification and prevents, for example, a relational table from owning improperly a OLAP Measure. In order to illustrate how CWM works, the object diagram in Fig. 2 is an example of a relational table being partially modeled by a CWM metamodel, where each object represents the table or a table element, as instance of the respective metamodel class.

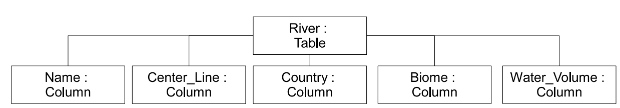


Fig. 2. CWM metamodel of a relational table

While geographic data are essential for many VA applications, the CWM metamodel does not specify specialized structures to represent them. So far there is one initiative to extend the CWM metamodel to afford geospatial types [8]. However, as will be further detailed, they do not consider the multi-representation dimension of geospatial data. In the context of GIS and cartography, the concept of generalization refers to the multiple representations a spatial phenomenon may assume, according to time and scale variations [9]. For example, at a 1:10,000 scale, one may represent a building (an abstract entity) by its edges (representation), meanwhile at a 1:100,000 the same building may be represented by a point (representation). An object can also vary in its representations, independently of scale and time, according to the abstraction level, users' points of view, or requirements of algorithms. In our model the representation of an object is a dimension, orthogonal to time and scale. For example, consider a set of temperature readings in a city: it's possible to represent them, in the same scale, by plotting the point-set, or by aggregating them in an isoline surface (see Fig. 3).

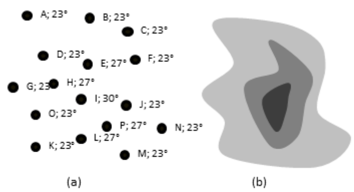


Fig. 3. Point set representing temperature sensors (a) and the generated isoline surface (b)

As mentioned before, our approach is designed to address VA systems. Through VA tools, people can synthesize information, derive insights from massive, dynamic and ambiguous data [1]. VA comprises multidisciplinary fields like: analytical reasoning, which enables users to obtain insights to support assessment, planning and decision-making; visual representation and interaction techniques, which allow users to explore and understand large amounts of information at once; production, presentation and dissemination of results.

## IV. Lens Approach

This work proposes an on the fly and on demand data unification approach, named Lens, founded in a multidimensional, multi-representation homogeneous data model, tailored to VA. The data metamodel was split in two sub-models: a Unifying Metamodel and an Analytics Metamodel. In the two following subsections, we present this data model and its architecture.

### A. Unifying Metamodel

A core characteristic in our Lens approach is the ability of unifying data from different Web sources on the fly. It involves not only to deal with different schemas, but also aligning distinct data models in a homogeneous one. Due to the diversity of models for data sources on the Web, we chose a representative subset to work with. But we designed our unifying approach to be extensible to other models.

By unification and homogenization we mean that our architecture will afford inputs represented in many data models and will map them to a single output model. We chose the multidimensional data model as the unifying one, since this architecture is tailored to VA operations, which are typically OLAP operations in the data access perspective.

Any data source to be integrated is described using CWM metamodels (either relational, object, etc). Then transformation rules are defined, in terms of CWM Transformation metamodel classes in order to map each element of the source schema to the target multidimensional output schema. Further details on how CWM Transformation is used are out of scope of this paper.

Since our multidimensional output data is also multi-represented, the output schema would hold not only conventional data, as a OLAP cube would, but also spatiotemporal (both raster and vector) in multiple aligned dimensions and representations. To afford spatiotemporal data and multi-representation, this work extends the CWM metamodels by adding new spatial data types, multi-representation features and the corresponding operations. In this sense, our metamodel unifies the geometry types defined in [10], [11], [4]. A representative subset of their data types was mapped into a common metamodel classes.

As illustrated in Fig. 4, we extended the core CWM type system to represent multi-represented objects based on two building blocks: the abstract characterization of an abstract object independent of its representation (`Spatial::AbstractSpatialObject`), and the concrete spatial representation of a given spatial object (`Spatial::SpatialObjectType`). Henceforth we will omit the `Spatial` namespace for simplicity reasons. By extending the core typing model, any metamodel (either relational, object, multidimensional, etc.) can refer to spatial representations and thus it is possible to map them to the Unifying Model.

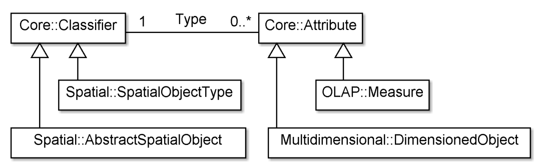


Fig. 4. Geometry and multi-representation extension to CWM data types

Fig. 5 shows an expansion of the class `SpatialObjectType` and its relations. It abstracts either a vector (points, lines), a field (triangulated irregular networks), or a network (a topological connection of nodes and arcs) representation. Fig. 6 shows how we achieve a multi-representation by relating the two building block classes: `AbstractSpatialObject` and `SpatialObjectType`. Each `AbstractSpatialObject` can be related to one or more alternative representations (`RepresentationType`), which in turn will be presented as a `SpatialObjectType`. As example, a river (`AbstractSpatialObject`) may assume different representations (a flood area, a center line, etc), related to instances of a `SpatialObjectType`.

### B. Analytics Metamodel

While the role of the Unifying Metamodel is to integrate and homogenize distinct data models, the Analytics Metamodel is concerned with modeling the VA and OLAP data derived from the Unifying Metamodel. Therefore, the Analytics Metamodel extends the CWM OLAP metamodel, being aligned to the Unifying Metamodel, which extends the CWM Multidimensional

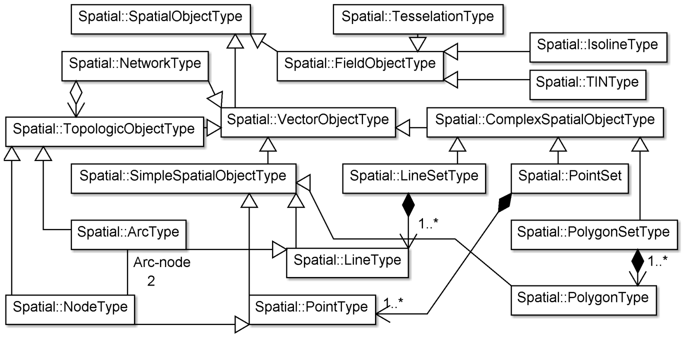


Fig. 5. Full inheritance tree of Spatial::SpatialObjectType

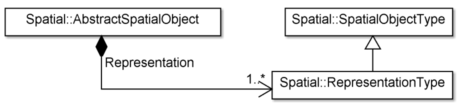


Fig. 6. Depiction of Spatial::AbstractSpatialObject

metamodel. Lens extends CWM OLAP hypercubes, adding multi-representation capabilities, i.e., allowing several data materializations of the same elements: they can be generalized (aggregated, transformed, combined, etc). For example, an equipotential surface can be generated from a set of points (see Fig. 3) and both representations will be kept in the hypercube. While conventional columns are summarized by using usual traditional OLAP operations, spatiotemporal columns require spatial operations, e.g. topological relation operators: intersects, overlaps, etc. Therefore, hypercubes require extra operations, modeled as specializations of CWM Transformation, for data materialization.

In order to add multi-representation capabilities, we specialized `Core::Classifier` class into `SpatialObjectType`, a spatiotemporal element, and `AbstractSpatialObject`, an abstract multi-representable object, as show in Fig. 4. The Measure class represents OLAP measures, as presented in Section III. Our extension turns a `Measure` into a spatiotemporal measure. Fig. 7 depicts an example where the River table from Fig. 2 was used as basis for a cube with two measures: `water_volume` which is a `Measure` and `RiverShapes`, that is also a measure, related to `RiverRepresentation`, which is an instance of `AbstractSpatialObject`. It is in turn associated with two different representations (refer from Fig. 4 to 6). The `CenterLine` representation is a `LineSetType`, as rivers may have many meanders, while `FloodArea` is a `PolygonType`.

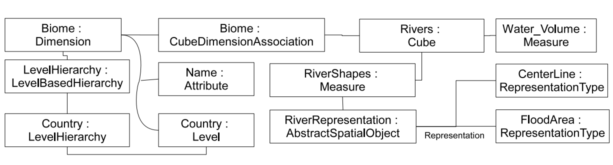


Fig. 7. CWM metamodel of Rivers Cube, enriched with multi-representation measure

The CWM OLAP metamodel is the basis for our analytics model due to many reasons. First, it describes logical analytical models but does not specify how these models should be deployed, providing freedom to choose the best storage structure (for example, serialized in plain files, star-schemas, BigTables), as well as transparency, although it is necessary to map instances of OLAP metaclasses to instances of other CWM metamodels. Second, as previously said, OLAP is a model appropriate for high performance analysis and retrieval.

This model subsidizes our storage strategy. Initially data is transformed to a multidimensional model and kept at the lowest level of granularity, which is application dependent. As data is processed and new representations are generated, the results are stored back into hypercubes. New representations generated during analytical operations can be stored at the same granularity level of the previous representation (horizontal materialization) or can be part of a representation at a coarser level of granularity (vertical materialization). Vertical materialization thus will append new lines to the hypercube, while horizontal materialization will append a new representation to the `AbstractSpatialObject` aggregated set. Transformation activities performed by clients (human or other systems), which lead from one representation to another, are modeled using a set of `TransformationActivity`, a class defined in the Transformation CWM metamodel [5]. It allows to keep track of different materializations, which analysts produce and store in hypercubes.

## V. Lens Architecture

The Lens metamodel is the basis to model a set of one or more multi-representation hypercubes, which are implemented as data structures, related to a set of components aimed to manage them. This proposed Lens Architecture aims at connecting to any data source, structured and unstructured, local or in the cloud, making data access transparent to users.

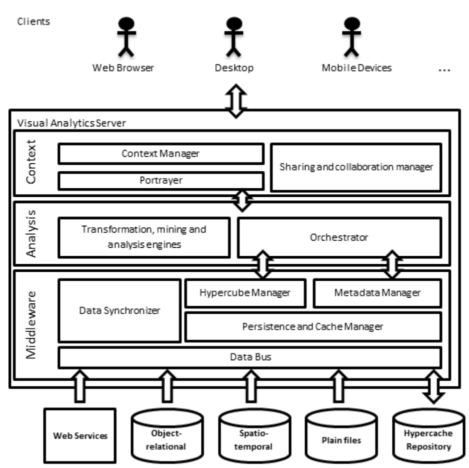


Fig. 8. The proposed Middleware and supporting architecture

The Lens is intended to be part of a larger architecture for VA applications. This architecture, shown in Fig. 8, was designed to fulfill the requirements proposed in Section II. It is composed of layers responsible for specific tasks. Each layer communicates with the layer immediately above or below, through a set of well-defined interfaces and services. Layer isolation allows transparency and, therefore, reduces coupling, while being structured as services allows distribution. The Lens, in the referred architecture, corresponds to the Middleware layer, whose components are: **Data Bus**, responsible for data source connection, thus abstracting original data repositories. Data sources are read-only. It is possible to add new data source types by writing new drivers and appending them to this component. New models are mapped by using Lens metamodels. The Data Bus is responsible to retrieve data and forward them to the Data Synchronizer. The data bus is also responsible for I/O operations in the hypercache repository, which stores the multi-representation hypercube. **Data Synchronizer** detects changes in source repositories and propagates them to the multi-representation hypercube. Updates are, ideally, made lazily (i.e., when the workload is low enough) and as needed, this way avoiding affecting performance. It also can be made in batch mode, e.g., when the hypercube is first created. When new data sources are registered, the Data Synchronizer will activate the Hypercube Manager in order to generate a new hypercube. **Persistence and Cache Manager**, responsible for cache managing and mediation of I/O operation to the data bus. **Hypercube Manager**, a key component, responsible for keeping the data structure that implements the models defined in Section IV-A, necessary for the multi-representation hypercube and its capabilities. It can store conventional, spatiotemporal and hypermedia data. It manages data necessary for analysis and visualization. Several hypercubes can be available for analysis and they can be combined by the manager. Data is accessed via query language, an MDX extension with support to multi-representation. **Metadata Manager** maintains metadata and ontologies related to stored data. They are manipulated by using an API and an appropriate language.

VA operations are performed by the Analysis Layer. It interacts with the Middleware Layer via the API provided by the Middleware. The Analysis Layer is composed by the following modules: **Transformation, Mining and Analysis Engines** is a series of engines designed to act on the hypercube, metadata and ontologies. New algorithms, heuristics, etc. can be implemented as plugins attached to this module. **Orchestrator** is responsible for synchronizing and pipelining engines and procedures invoked by users, thus reducing coupling among the engines. The Orchestrator will interact with the Hypercube Manager using an API and query language.

The Context layer, which is responsible for users interaction, is composed by the following elements: **Context Manager** manages dashboards and cockpits users employ in their analysis. It contains a series of controls and interface elements (tables, maps, graphics, etc.) users compose in order to analyze data. Ideally, the context manager is client-aware: web browses running in tablets or smartphones do not have the same capabilities of fat clients and workstations, and it must be met by the context manager. **Portrayer** is responsible for setting the appropriate visual representation in GUI (graphical user interfaces) for data and analysis for the intended client. Graphics, graphs, tables, images, maps, etc. are generated and forwarded to the Context Manager. User inputs, captured by the Context Manager, are analogously forwarded to the orchestrator for treatment. **Sharing and Collaboration Manager** responsible for managing, for each user, the data and analysis that are shared with collaborators.

## VI. Related Work

There are many works concerning data models for multi-representation. Parent et al. [10] propose a entity-relationship data model, named MADS, comprising data structures for multi-representation of space and time. This data model is materialized into a framework named MurMur. Zhou and Jones [2] proposes a multi-representation model, named Multi-representation geometry (Mrep-Geometry), concerned with cartographic generalization. The VUEL concept [4] focuses on combining semantics, geographic and graphic representation of spatial elements. These works rely on a centralized and monolithic database storage, based on a single model, while Lens rely on a middleware-based integration architecture, founded in an expansible unification metamodel to transparently extend any underlying model. None of these works are concerned with high performance queries, demanded by VA, as their approach relies on an object-relational model.

Many initiatives address the combination of OLAP with spatiotemporal and hypermedia capabilities, namely hypermedia hypercubes. However, we have not found so far any work addressing multi-representation on multidimensional models, as proposed in this work. Han et al [12] propose spatial data warehouses based on spatial data cubes, mixing spatial and non-spatial measures. Papadias et al. [13] proposes structures for ad-hoc group-by queries on spatial data in star-schema structures. In [8] the authors propose a metamodel to add geographical dimensional schemas, based on the OLAP CWM [6] metamodel. [14] present many possibilities to explore spatial data in OLAP cubes. The work [15] combines hypermedia documents and spatial data in OLAP hypercubes (hypermedia hypercubes). Piet is an implementation of a GIS-OLAP integration [16] presenting a method for precomputation of spatial overlay aggregations. The GooLAP [17] proposes a three-tier architecture to build a system for spatial OLAP operations; but the authors split data in two different databases, a geographic boundaries database and a spatial data warehouse, while our approach unifies shapes in a centralized data repository.

Concerning the architecture, there are many different approaches for a VA applications, from highly coupled and centralized, to loosely coupled and distributed. All analyzed proposals rely on a central repository representing data in relational or object-relational models, without support to multi-representation. For example, [18] presents a component-based architecture, relying on a central repository for time data. Sunfall [19] is a VA system for collaborative works on astrophysics, for applications demanding high throughput of astronomic imagery data processing, built around a central repository. Finally SQuAVisiT [20] is a plugin based framework, centered on a repository, which receives, forwards and stores data that is used by analysts.

Relational multi-representation models, like MADS, are very good for storing elements for transaction processing and for representing complex relationships; on the other hand, spatial OLAP cubes, like [17], are apt to queries, but lack the flexibility for representing and track different perspectives of data. OMT-G [11], on the other hand, considers multi-representation only on modeling level. Our proposal combines the power of OLAP cubes and multi-representation in an effort to support fast and flexible queries for VA applications. The hypercube model is a consolidated approach to speed up queries. Moreover, approaches relying in a central repository, e.g., MurMur and Piet, will hardly address requirements of accessing distributed and on demand data, usual in VA applications. Therefore, this work adopted a middleware-based approach instead of using a centralized database. Among the reasons that lead us to this option, we point out: first, there are a great number of database management systems available, both free and commercial, and a great amount of data already available in these databases, as well as data scattered in files, images, etc. The Lens middleware raises the abstraction level, keeping data where they are, with no modification or migration impacts. Copies of data are consolidated transparently by the middleware, which remains responsible for detecting changes.

## VII. Conclusions and Future Work

This work presented Lens, a combination of a unification metamodeling approach apt to support multi-representation and multidimensional data, and a middleware based architecture, which allows users to dynamically build perspectives of data, on the fly and on demand, for VA applications. Lens supports VA by allowing transformation, aggregation and analysis of massive, heterogeneous, distributed data. Lens will transparently and seamlessly allow users to fetch data from their original sources and build multi-representation hypercubes, managing metadata, as well as providing an API and a query language for manipulating the hypercubes.

Future work includes the implementation of the proposed architecture and will tackle the following problems: the application of interoperable semantics to data and representations; the design of efficient and intelligent synchronization mechanisms (deciding when to automatically update hypercubes or not); the application of efficient mechanisms to deal with sparse data, and multi-representation hypercubes building on the fly with good performance.

## Acknowledgment

This work was partially financed by CNPq, FAPESP, CAPES-COFECUB (AMIB project) and INCT in Web Science (CNPq 557.128/2009-9).

## References

[1] J. T. Thomaz and K. A. Cook, Eds., *Iluminating the Path: The research and development agenda for Visual Analytics*. National Visualization and Analytics Center, 2004.

[2] S. Zhou and C. B. Jones, "A Multi-representation Spatial Data Model," *Lecture Notes in Computer Science*, vol. 2750, pp. 394–411, 2003.

[3] C. Vangenot, "Multi-representation in spatial databases using the MADS conceptual model," in *ICA Workshop on Generalisation and Multiple Representation*, no. August, Leicester, 2004, pp. 1–8. [Online]. Available: http://ica.ign.fr/Leicester/paper/Vangenot-v2-ICAWorkshop.pdf

[4] Y. Bédard, E. Bernier, and T. Badard, "Multiple representation spatial databases and the concept of Vuel," *Encyclopaedia in Geoinformatics, Hershey: Idea Group Publishing, accepted*, vol. 2131, no. 7116, 2007.

[5] OMG, *Meta Object Facility (MOF) Core Specification Version 2.0*, 2006.

[6] D. Chang and S. Iyengar, "Common warehouse metamodel (CWM) specification," *Object Management Group*, vol. 1, no. March, 2001.

[7] J. Poole, D. Chang, D. Tolbert, and D. Mellor, *Common Warehouse Metamodel Developer's Guide*. Wiley, 2003.

[8] R. N. Fidalgo, V. C. Times, J. Silva, F. F. Souza, and A. C. Salgado, "Providing Multidimensional and Geographical Integration Based on a GDW and Metamodels," in *Proceedings of the ISPRS Joint Workshop on Spatial, Temporal and Multi-Dimensional Data Modelling and Analysis*, Quebec, Canada, 2003.

[9] W. Mackaness, A. Ruas, and L. T. Sarjakoski, Eds., *Generalisation of Geographic Information Cartographic Modelling and Applications*. Elsevier, 2007.

[10] C. Parent, S. Spaccapietra, E. Zimanyi, P. Donini, C. Plazanet, and C. Vangenot, "Modeling Spatial Data in the MADS Conceptual Model," *Symposium A Quarterly Journal In Modern Foreign Literatures*, pp. 1–12, 1998.

[11] K. Borges, C. Davis, and A. Laender, "OMT-G: an object-oriented data model for geographic applications," *GeoInformatica*, vol. 5, no. 3, pp. 221–260, 2001.

[12] J. Han, N. Stefanovic, and K. Koperski, "Selective materialization: An efficient method for spatial data cube construction," in *Research and Development in Knowledge Discovery and Data Mining*, ser. Lecture Notes in Computer Science, X. Wu, R. Kotagiri, and K. Korb, Eds. Springer Berlin / Heidelberg, 1998, vol. 1394, pp. 144–158.

[13] D. Papadias, P. Kalnis, J. Zhang, and Y. Tao, "Efficient OLAP operations in spatial data warehouses," in *Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases*. Springer, 2001, pp. 443–459. [Online]. Available: http://www.springerlink.com/index/DWH7T0720YLFVTM0.pdf

[14] S. Rivest, Y. Bedard, M. Proulx, M. Nadeau, F. Hubert, and J. Pastor, "SOLAP technology: Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, no. 1, pp. 17–33, Dec. 2005.

[15] Y. Bédard, M.-J. Proulx, S. Rivest, and T. Badard, "Merging Hypermedia GIS with Spatial On-Line Analytical Processing: Towards Hypermedia SOLAP," *Lecture Notes in Geoinformation and Cartography*, 2006.

[16] B. Kuijpers and A. Vaisman, "A data model for moving objects supporting aggregation," in *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*. IEEE, 2007, pp. 546–554.

[17] S. Di Martino, S. Bimonte, M. Bertolotto, F. Ferrucci, and V. Leano, "Spatial OnLine Analytical Processing of Geographic Data through the Google Earth Interface," *Geocomputation, Sustainability and Environmental Planning*, pp. 163–182, 2011. [Online]. Available: http://www.springerlink.com/index/0676Q72026212738.pdf

[18] W. Aigner, A. Bertone, S. Miksch, C. Tominski, and H. Schumann, "Towards a conceptual framework for visual analytics of time and time-oriented data," in *Simulation Conference, 2007 Winter*, 2007, pp. 721–729.

[19] C. Aragon, S. Poon, G. Aldering, R. Thomas, and R. Quimby, "Using visual analytics to maintain situation awareness in astrophysics," in *Visual Analytics Science and Technology, 2008. VAST'08. IEEE Symposium on*. IEEE, 2008, pp. 27–34.

[20] M. van den Brand, S. Roubtsov, and A. Serebrenik, "SQuAVisiT: A Flexible Tool for Visual Software Analytics," in *Software Maintenance and Reengineering, 2009. CSMR '09. 13th European Conference on*, 2009, pp. 331–332.