Temporal GIS and Spatiotemporal Data Sources

Karine Reis Ferreira, André Gomes de Oliveira, Antônio Miguel Vieira Monteiro, Diego Benincasa F. C. de Almeida

Image Processing Division (DPI)
National Institute for Space Research (INPE) - São José dos Campos – Brazil

karine@dpi.inpe.br, andre.oliveira@funcate.org.br, miguel@dpi.inpe.br, benincasa@dpi.inpe.be

Abstract. *The recent technological advances in geospatial data collection have created massive data sets with better spatial and temporal resolution than ever. To properly deal with these data sets, geographical information systems (GIS) must evolve to represent, access, analyze and visualize big spatiotemporal data in an efficient and integrated way. In this paper, we highlight challenges in temporal GIS development and present a proposal to overcome one of them: how to access spatiotemporal data sets from distinct kinds of data sources. Our approach uses Semantic Web techniques and is based on a data model that takes observations as basic units to represent spatiotemporal information from different application domains. We define a RDF vocabulary for describing data sources that store or provide spatiotemporal observations.*

1. Introduction

The recent technological advances in geospatial data collection, such as Earth observation and GPS satellites, have created massive data sets with better spatial and temporal resolution than ever. This scenario has motivated a challenge for Geoinformatics. We need geographical information systems (GIS) able to deal with big spatiotemporal data sets in an efficient and integrated way.

We use the term "temporal GIS" to refer to GIS that can model, access, combine, process, analyze and visualize spatiotemporal information. In the literature, there are many proposals of conceptual models to represent and handle spatiotemporal data in GIS and database systems. However, there is not yet a full-scale and comprehensive temporal GIS available (Yuan, 2009). Most existing temporal GIS technologies either are still in the research phase or are specific for certain application domain.

In this paper, we highlight challenges faced in temporal GIS development and present a proposal to overcome one of them: *how to access spatiotemporal data sets from distinct kinds of data sources*. Our approach uses Semantic Web techniques and is based on a data model that takes observations as basic units to represent spatiotemporal information from different application domains. We define a RDF vocabulary to describe spatiotemporal data sources. A preliminary proposal of this vocabulary was described as ongoing work in Ferreira et al (2014).

RDF (Resource Description Framework) is a data model for describing and connecting resources and SPARQL is a query language for RDF data sets. Both are World Wide Web Consortium (W3C) standards and are key techniques in Linked Data and Semantic Web (Berners-Lee et al, 2001). RDF describes resources using the concepts of classes, properties, and values. The term vocabulary refers to a set of classes and properties that are defined specifically for a certain application.

1.1. Related work

In recent years, traditional GIS technologies have being extended to deal with spatiotemporal data. Gebbert and Pebesma (2014) present a field based temporal GIS, called TGRASS, based on the open source Geographic Resources Analysis Support System (GRASS). This system works with *fields* or *coverages*, following the definition of Galton (2004): *a spatial field is a mapping from spatial locations to values that may be any kind of data structures*.

Time Manager[1] is a plugin of the open source system QGIS (Quantum GIS). This plugin provides tools, such as a time slider, that allow users to animate vector and raster layers based on time attributes. It focuses on visualization of temporal layers, creating animations directly in the map window and exporting image series. Tracking Analyst is a module of the commercial ArcGIS software (ESRI, 2010). Using this module, users can add temporal data to ArcGIS, visualize it dynamically and handle it as trajectories of objects.

Peuquet and Hardisty (2010) present a web-based tool, called STempo, for visualizing and analyzing space-time event data sets and for discovering patterns from these sets. STEMgis[2] is a commercial temporal GIS to dynamically visualize and explore spatial data throughout the four dimensions of space and time.

Following this trend, we are extending the TerraLib GIS library and TerraView GIS software (Camara et al, 2008) to deal with spatiotemporal data. Differently from the previous systems cited in this section, this extension does not focus on a specific type of spatiotemporal data. TGRASS works with fields or coverages, Tracking Analyst with trajectories of objects and STempo with space-time events. Our goal is to develop a more comprehensive temporal GIS able to work with fields, trajectories as well as events. Many applications need temporal GIS that can integrate different types of spatiotemporal data.

2. Spatiotemporal data and applications

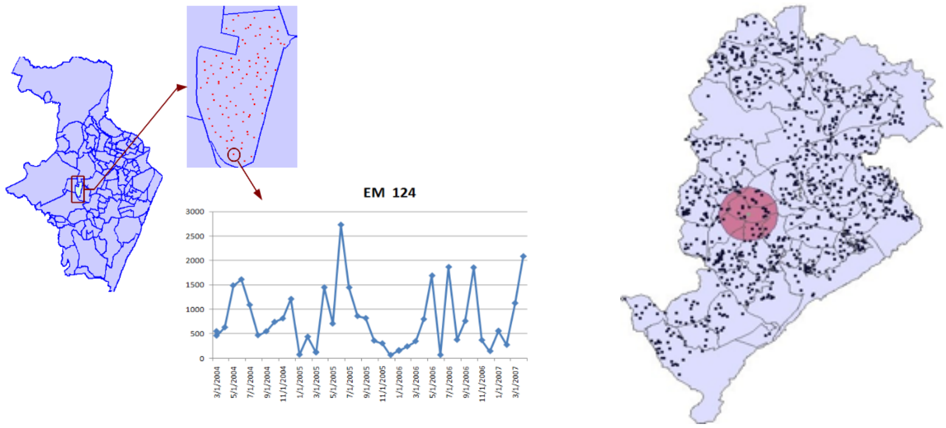
This section aims at showing the diversity of spatiotemporal data from different application domains. We present real examples of data sets from public health, location-based services, environmental and natural disaster monitoring applications.

Figure 1 (a) shows time series used in disease surveillance of dengue in the city of Recife, Brazil (Regis et al, 2009). Dengue is a viral disease transmitted by the *Aedes aegypti* mosquitoes. These mosquitoes lay their eggs in standing water; the eggs hatch in hot weather. To assess dengue risk, health services use buckets of water as egg traps. Each trap has a fixed location represented as a red point in the picture. The time series represents the number of mosquito eggs gathered weekly from an egg trap in a district

[1] Available at: http://anitagraser.com/projects/time-manager/

[2] Available at: http://www.discoverysoftware.co.uk/STEMgis.htm

of Recife. Figure 1 (b) presents occurrences of meningitis in Belo Horizonte city. Each event has a spatial location (black points in the picture) and a time of occurrence.



(a) **(b)**

Figure 1. (a) Number of mosquito eggs gathered from an egg trap in Recife, Brazil; and (b) Occurrences of meningitis in Belo Horizonte city.

Figure 2 presents routes of eight sea elephants in Antarctica. These animals were monitored by a project called MEOP - "Marine Mammal Exploring the Oceans Pole to Pole" (http://www.inpe.br/crs/pan/pesquisas/telemetria.php). The project monitored the trajectories of these animals during 3 years, shown as red lines in the figure.

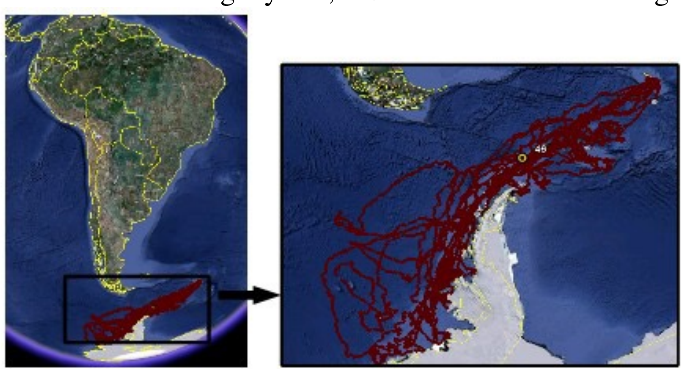


Figure 2. Trajectories of eight sea elephants in Antarctica (red lines).

Figure 3 shows a set of observations (red points) gathered in a lake of Amazon rainforest in two different months. Each observation measures the chlorophyll value, among other properties, at a specific location and time. These observations are taken monthly to analyze the variation of chlorophyll within the lake over time. Usually, a kriging interpolation function is used to estimate values at non-observed locations.

Figure 4 shows two grids; each one associated to a time. These grids contain the rain variation in the state of Rio de Janeiro during the natural disaster of 11 January 2011. Each cell contains an estimated value of precipitation, in millimeter per hour (mm/h). These grids are taken in 15-minute intervals.

To meet demands from different application, a temporal GIS has to deal with distinct types of spatiotemporal data in an integrated way. This includes: (1) time series associated to fixed spatial locations (Figure 1(a)); (2) events or occurrences that happen in a certain time and space (Figure 1(b)); (3) trajectories of moving objects (Figure 2);

(4) fields or coverages based on measures associated to spatial locations and times and on interpolation functions (Figure 3); and (5) sequence of raster or grid data over time (Figure 4).

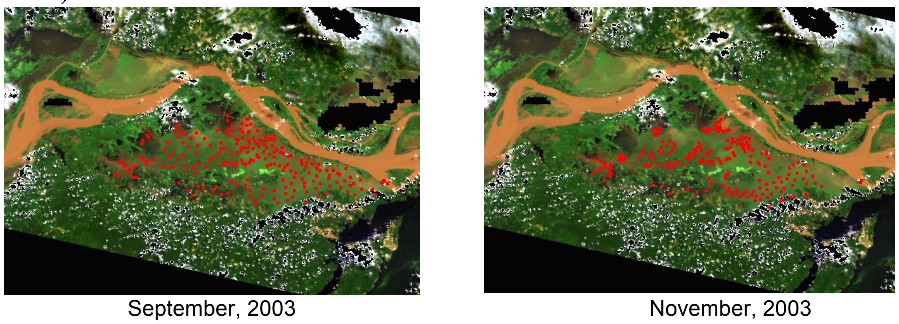


Figure 3. Observations of chlorophyll in a lake of Amazon rainforest.

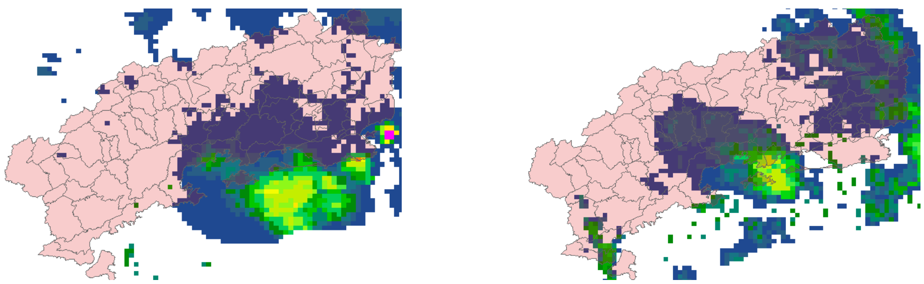


Figure 4. Precipitation grids in the state of Rio de Janeiro, Brazil, in 11 January 2011.

Each type of spatiotemporal data requires a specific set of operations. For example, we can use an interpolation function associated to Amazon lake observations in order to estimate chlorophyll values in non-measured locations and times. However, this operation does not make sense for meningitis occurrences. Besides that, the algorithm to find spatiotemporal clusters for meningitis occurrences is different from the one for trajectories. The first algorithm is suitable for independent events (Veloso, 2013) while the second has to taken into account the association between trajectories and objects.

Moreover, a temporal GIS must be able to access different kinds of spatiotemporal data sources. For instance, the egg trap time series (Figure 1(a)) are stored in a PostGIS database; the meningitis occurrences (Figure 1(b)) are available through Web Feature Service (WFS); the sea elephants trajectories (Figure 2) are stored in a Keyhole Markup Language (KML) file; the Amazon lake observations (Figure 3) are available as a set of Shapefiles and each file is associated to a specific time; and the precipitation grids (Figure 4) are stored as Geotiff files and each file is associated to a specific time.

3. Temporal GIS development

To organize the discussion about challenges in temporal GIS development, we consider a general architecture composed of four modules, as shown in Figure 5. The *Data Model* module is the core of the system. It defines the key concepts of the system that reflect on all other modules. This module contains data types, relationships, operations and rules to represent these concepts. *Data Access* is a module responsible for accessing and querying data sets from different kinds of sources and for mapping these data sets into the concepts and types of the *Data Model*.

Data Processing and Analysis module contains methods and algorithms to process and analyze data. In most systems, users can directly interact with this module through script languages, such as Python (www.phython.org) and LUA (www.lua.org). Script languages allow users to express complex processing. *Data Presentation* module contains all elements associated to data visualization, including the system Graphical User Interfaces (GUI). It is responsible for data presentation to the user. In this paper, we focus on the two first modules, *Data Model* and *Data Access*.

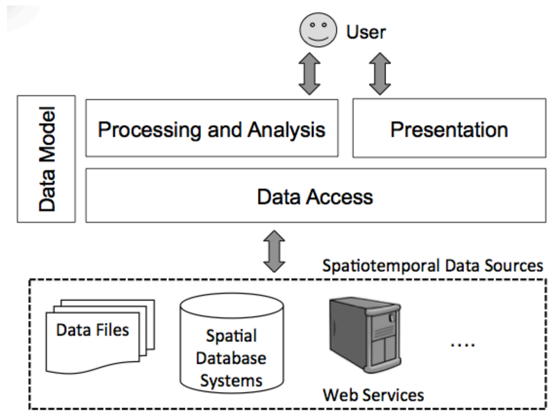


Figure 5. GIS general architecture.

3.1. Temporal GIS: Data Model

The first challenge faced in temporal GIS implementation is: *how to model spatiotemporal data*. We need a data model that defines a minimal set of data types able to represent different kinds of spatiotemporal information from distinct application domains.

In GIScience, static geospatial information is represented following well-established models and concepts. This includes the dichotomy between *object-based* and *field-based* models (Galton, 2004). Examples of long-standing concepts are vector and raster data structures, topological operators, spatial indexing, and spatial joins (Rigaux et al, 2002). Most existing GIS and spatial database systems, such as PostGIS and Oracle Spatial, are grounded on these concepts. However, there is no consensus on how to represent spatiotemporal information in computational systems.

Many existing proposals of spatiotemporal data models focus on representing the evolution of *objects* and *fields* over time. Pelekis et al. (2004) review some of these models and consider that most of them are data-specific; each one addresses a class of spatiotemporal data. Some proposals are specific for discrete changes in objects (Worboys, 1994) (Hornsby and Egenhofer, 2000), others for moving objects (Güting and Schneider, 2005) (ISO, 2008) and still others for fields or coverage (Liu et al, 2008) (OGC, 2006). However, many applications need to combine different classes of such data. For example, environmental change and natural disaster monitoring have to deal with moving objects as well as with fields. Thus, we need spatiotemporal data models as generic as possible to support such applications.

To properly capture changes in the world, representing evolution of objects and fields over time is not enough. We also need to represent events and relationships between events and objects explicitly (Worboys, 2005). Events are *occurrents* (Galton and Mizoguchi, 2009). They are individual happenings with definite beginnings and ends. The demand for models that describe events has encouraged recent research on spatiotemporal data modeling (Worboys, 2005) (Galton and Mizoguchi, 2009).

3.2. Our proposal: An Observation-Based Model for Spatiotemporal Data

We proposed a data model for spatiotemporal data and specified it using an algebraic formalism. Details about this data model and its algebra are in (Ferreira et al, 2014). Algebras describe data types and their operations in a formal way, independently of programming languages. The proposed algebra is extensible, defining data types as building blocks for other types, as shown in Figure 6.

Observations are our means to assess spatiotemporal phenomena in the real world. Recent research draws attention to the importance of using observations as a basis for designing geospatial applications (Kuhn, 2009). The proposed model takes observations as basic units for spatiotemporal data representation and allows users to create different views on the same observation set, meeting application needs.

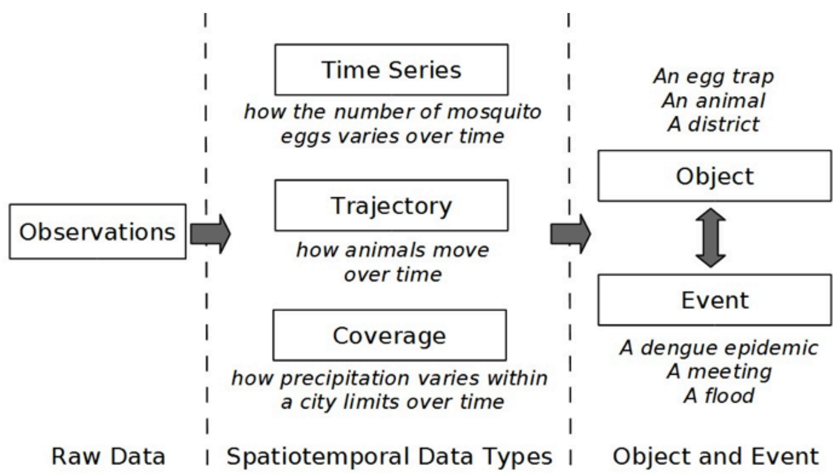


Figure 6. An Algebra for Spatiotemporal Data: From Observations To Events. Source: (Ferreira et al, 2014)

We define three spatiotemporal data types as abstractions built on *observations*: *time series*, *trajectory*, and *coverage*. A *time series* represents the variation of a property over time in a fixed location. A *trajectory* represents how locations or boundaries of an object change over time. A *coverage* represents the variation of a property in a spatial extent at a time. We also define an auxiliary type called *coverage series* that represents a time-ordered set of coverages that have the same boundary. Using these types, we can represent objects and fields that change over time as well as *events*.

We implemented all data types and operations of the proposed algebra in a new module called "TerraLib ST" in the TerraLib library. Using this module and its data types, we can represent and combine different kinds of spatiotemporal data sets from distinct application domains, such as the ones presented in section 2.

3.3. Temporal GIS: Data Access

The second challenge faced in temporal GIS implementation is: *how to access and integrate spatiotemporal data sets from different kinds of data sources*. There are not standards on how to store spatiotemporal data in spatial database systems or files as well as on how to serve such data through web services.

Since the beginning of the 2000s, the GIS community has made a serious effort towards spatial data interoperability. The International Organization for Standardization (ISO) and the Open Geospatial Consortium (OGC) have proposed standards3 to represent and store spatial information in data files and database systems as well as to serve spatial data via web services. Geography Markup Language (GML) and KML are examples of data formats proposed by OGC for spatial data interchange. Spatial extensions of traditional object-relational Database Management Systems (Spatial DBMS), such as PostGIS and Oracle Spatial, deal with vector spatial information in compliance with the OGC Simple Feature Access (SFA) specification. Regarding web services, there are standards for serving spatial data, metadata and processes, such as Web Feature Service (WFS), Web Coverage Service (WCS), Catalogue Service Web (CSW) and Web Processing Service (WPS).

The compliance with ISO and OGC standards has assured a high degree of spatial data interoperability. Many GIS tools and libraries are able to access spatial data files, databases and web services that follow these specifications. Standards are useful to promote spatial data interoperability. However, few results have been achieved regarding spatiotemporal data interoperability. Most OGC and ISO standards are related to spatial but not spatiotemporal data.

Regarding spatiotemporal data, OGC proposes a standard called Sensor Observation Service (SOS) that defines a web service interface for disseminating and querying spatiotemporal observations, sensor metadata and observed features, based on the OGC Observations and Measurements (O&M) specification (OGC, 2010). However, many data providers store and disseminate spatiotemporal information using other formats and standards, not only SOS. GIS tools must be able to access different types of spatiotemporal data sources, without forcing the use of a specific format or standard.

3.4. Our proposal: A RDF vocabulary for spatiotemporal data sources

We consider that data sources store and provide spatiotemporal observations, which are basic units for spatiotemporal data representation. A temporal GIS must access these observations from data sources and allow users to create different views on them, according to application needs.

We propose an approach to access spatiotemporal observations from different kinds of data sources using Semantic Web techniques. The central idea is to use RDF files for describing *how* spatiotemporal observations are stored and provided by data sources and SPARQL language for discovering information about these data sources. We use RDF as linked metadata files, that is, files that describe *how* data sources represent spatiotemporal observations and *links* among these data sources. We do not

[3] http://www.opengeospatial.org/standards/is

transform the spatiotemporal observations from their original data sources and formats into RDF files. Each data source has an associated RDF file and all RDF files are based on the same vocabulary, as presented in Figure 7.

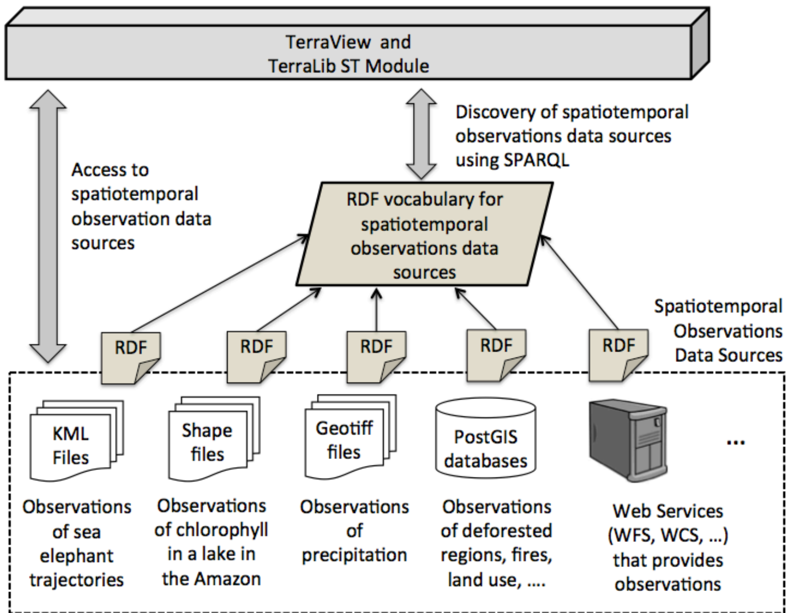


Figure 7. A proposal to access different kinds of spatiotemporal data sources.

There are many RDF vocabularies for different application domains, such as Dublin Core[4] for describing documents and FOAF[5] (Friend Of A Friend) for describing relationships among people. Examples of RDF vocabularies for describing geospatial data are W3C Basic Geo vocabulary, GeoOWL ontology, NeoGeo Vocabulary and GeoSPARQL (Battle and Kolas, 2012). GeoSPARQL is an OGC standard that defines a vocabulary for representing geospatial data in RDF and an extension to the SPARQL query language for processing geospatial data.

We define a RDF vocabulary for describing spatiotemporal observation data sources, based on OGC O&M specification, OGC GeoSPARQL vocabulary for spatial data and ISO standards for time representation (ISO, 2002) (ISO, 2004).

4. A RDF vocabulary for spatiotemporal data sources

The proposed vocabulary is described using OWL (Ontology Web Language). In this paper, we use a UML class diagram to better represent the concepts of the vocabulary, as shown in Figure 8.

A data source can have one or more spatiotemporal observation data sets. The class `STODataSourceInfo` contains information about a data source (class `DataSourceInfo`) and its spatiotemporal observation data sets (class `STODataSetInfo`). There are three types of data sources: files (class `FileDataSourceInfo`), DBMS (class `DBMSDataSourceInfo`) and web

[4] http://dublincore.org/

[5] http://xmlns.com/foaf/spec/

services (class `WSDataSourceInfo`). Each type of data source is described by a specific set of attributes. For example, to describe a DBMS data source, we have to inform its host name (`host`), port number (`port`), database name (`db_name`), a user (`user`) and its password (`password`). To describe a data source composed of files, we just need the path (`path`) of the folder where the files are. In this first version, the vocabulary accepts three types of files (shapefile, Geotiff and KML), one type of DBMS (PostGIS) and one type of web services (WFS). They are described in the enumeration `DataSourceType`.

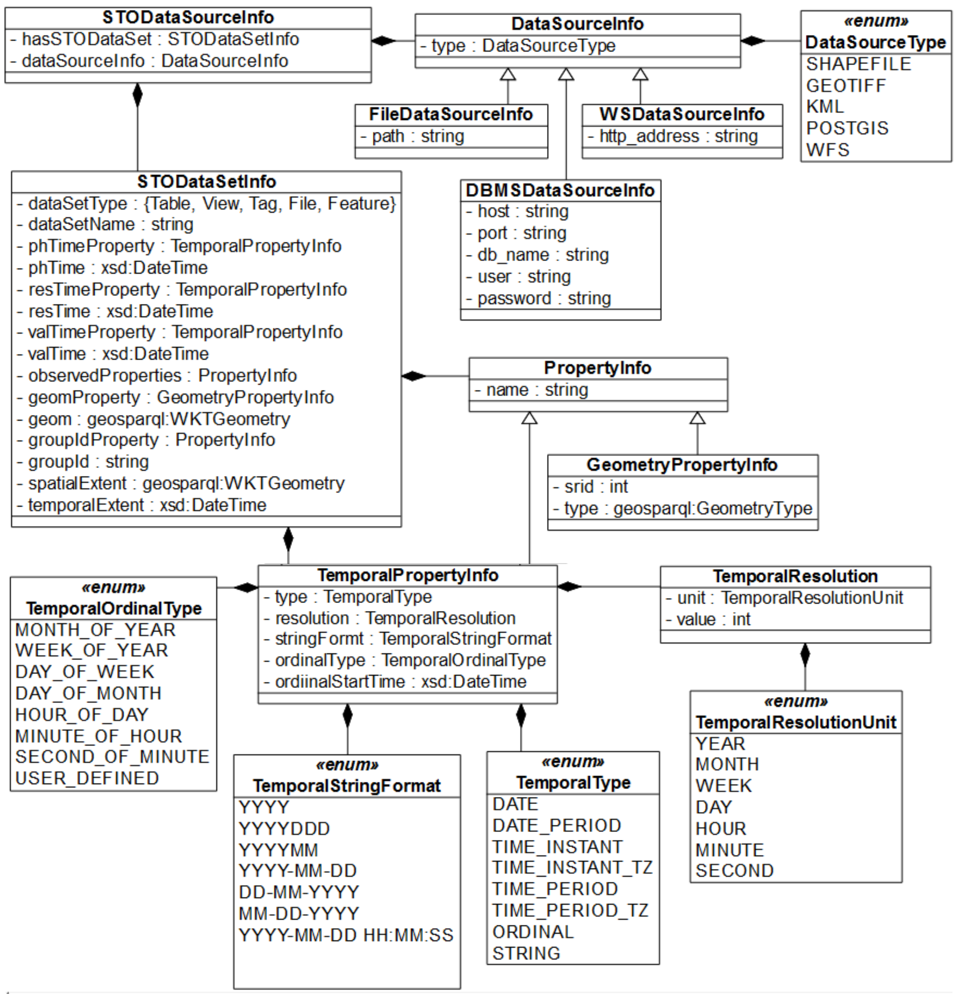


Figure 10. The RDF vocabulary for spatiotemporal observation data sources.

The class `STODataSetInfo` has information about data sets that contain spatiotemporal observations. Its two first attributes are the data set type (`dataSetType`) and name (`dataSetName`). In DBMS data sources, a data set type can be a table (`Table`) or a view (`View`); in files data sources, it can be an internal file tag (`Tag`) or a file (`File`); and in web services data sources, it is a Feature (`Feature`). If data set type is `Table`, the data set name is the table name; if type is `View`, it is the view name; if type is `Tag`, it is the tag name, and so on.

According to O&M specification (OGC, 2010), an observation has three temporal properties: *phenomenon*, *valid* and *result* time. The *phenomenon time* represents the time, instant or period, when the observation is actually measured. The *valid* time describes the time period during which the observation is available to be used. The *result* time represents the instant when the observation becomes available, typically when the observed values must be processed before being used. In the vocabulary, the phenomenon, valid and result time can be stored in data set properties (`phTimeProperty`, `valTimeProperty` and `resTimeProperty`) or be informed by the user (`phTime`, `valTime` and `resTime`).

To describe information about data set properties that store time, we define the class `TemporalPropertyInfo`. A temporal property has:

- A name (`name` of the super class `PropertyInfo`).
- A type (`type`): the types are defined in the enumeration `TemporalType` based on the ISO 19108:2002 standard (ISO, 2002). A temporal property can store instants (`DATE`, `TIME_INSTANT`, `TIME_INSTANT_TZ`) or periods (`DATE_PERIOD`, `TIME_PERIOD`, `TIME_PERIOD_TZ`) of time. Besides that, it can store times as texts (`STRING`) or as integers that indicate ordinal times (`ORDINAL`).
- A temporal resolution (`resolution` attribute represented by the class `TemporalResolution`): it indicates the time granularity that must be considered to deal with the temporal property. For example, the precipitation grids, shown in Figure 4, are taken at each 15 minutes. So, its temporal resolution unit is `MINUTE` and value is 15.
- A string format (`stringFormat`): this is necessary when the temporal property uses a textual representation of time, that is, its type is `STRING`. In this case, we need to inform what format the string is. For example, the text "01-03-2008" is ambiguous; it can represent the first day of March in 2008 or the third day of January in 2008. So, we have to inform what format it follows in order to understand its right meaning. ISO 8601:2004 (ISO, 2004) proposes some date and time format representations, such as DD-MM-YYYY or MM-DD-YYYY, and we define the enumeration `TemporalStringFormat` based on them.
- An ordinal type (`ordinalType`): this information is necessary when the temporal property stores ordinal times, that is, its type is `ORDINAL`. ISO 8601:2004 (ISO, 2004) defines some ordinal times and we define the enumeration `TemporalOrdinalType` based on them. For example, the ordinal day numbers in the week (`DAY_OF_WEEK`) mean: 1 is Monday, 2 is Tuesday, 3 is Wednesday, and so on.
- An ordinal start time (`ordinalStartTime`): this information is only necessary when the temporal property type is `ORDINAL` and the ordinal type is `USER_DEFINED`. In this case, the temporal property has user-defined ordinal numbers and so the user must inform what date and time is related to the first ordinal number.

Besides phenomenon, valid and result times, a spatiotemporal observation data set must have one or more properties that store the observed values. Users must indicate these properties through the attribute `observedProperties` in the class `STODataSetInfo`. Spatiotemporal observations are associated to spatial locations.

We use geometry types, such as point and polygon, to represent such locations. The geometries associated to observations can be stored in a data set property (`geomProperty`) or be informed by the user (`geom`). We define the class `GeometryPropertyInfo` to contain information about the property that stores geometries: its name (`name` of the super class `PropertyInfo`), its spatial reference system identifier (`srid`) and its type (`type`) whose values come from the OGC GeoSPARQL vocabulary. We also use the `WKTGeometry` type of the GeoSPARQL vocabulary to represent geometry (`geom`).

A spatiotemporal observation data set can have a property that stores identifiers that are used to group observations. Users can indicate this property through the attribute `groupIdProperty` in the class `STODataSetInfo`. Otherwise, users can give an identifier associated to all observations (`groupId`). Finally, users can inform the spatial (`spatialExtent`) and temporal (`temporalExtent`) extents of all observations. If users do not inform these extents, they are calculated taking into account all observations.

4.1. Example

To illustrate the use of the RDF vocabulary to describe spatiotemporal observation data sources, lets take the time series shown in Figure 1 (a). All measures of the mosquito egg traps are stored in a table called "`measures`" of a PostGIS database called "`dengue_surveillance`". Each row of this table is an observation in a certain time associated to an egg trap. Figure 11 shows a subset of this table and a simplified version of the RDF file to describe it, based on the vocabulary.

The RDF file has information about the PostGIS database "`dengue_surveillance`" in the tag `<DBMSDataSourceInfo>`, such as its host, port number, database name, user and password. Besides that, this file has information about the data set that contains spatiotemporal observations in the tag `<STODataSetInfo>`. In this example, the data set is the table "`measures`" indicated in tags `<dataSetType>` and `<dataSetName>`. The observation phenomenon times are stored in the table property "`m_date`" whose type is `DATE`. The measures were collected weekly. Thus its temporal resolution unit is WEEK and the value is 1. They are indicated in the tag `<phTimeProperty>`. In this example, there are not result and valid times associated to observations.

The table property "`m_number_eggs`" stores the observed values, that is, the number of mosquito eggs, as informed in tag `<observedProperty>`. The table property "`m_trap_location`" stores the spatial location of the traps and so is indicated in tag `<geomProperty>`. The identifier of each trap is stored in the table property "`m_trap_id`". All observations associated to a trap have the same trap identifier. So, this property can be used to group the observations by trap. We inform this through the tag `<groupIdProperty>`.

As presented in Figure 11, the TerraLib ST Module has two parts: (1) STDataModel that contains all data types and operations to represent spatiotemporal information based on the proposed algebra presented in section 3.2; and (2) STDataLoader that contains methods and functions to interpret RDF files that describe spatiotemporal observation data sources, to access such observations and to map them into the data types of the STDataModel.

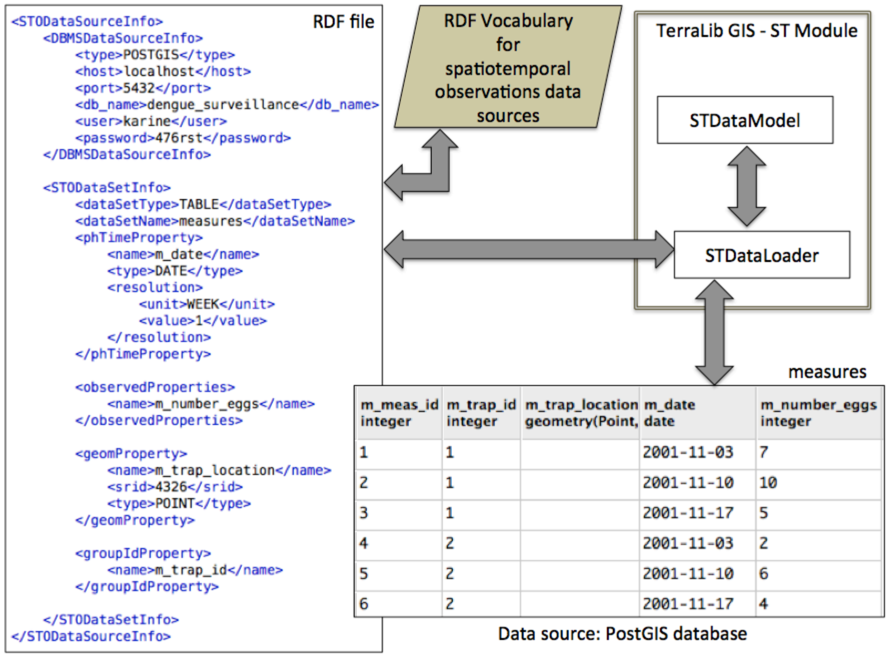


Figure 11. Example of the proposed RDF vocabulary.

5. Final remarks

This paper highlights challenges in temporal GIS development and presents a proposal to access spatiotemporal information from distinct kinds of data sources, using Semantic Web techniques. This approach consists in describing *how* data sources store spatiotemporal observations, based on a RDF vocabulary. Based on this description, temporal GIS are able to properly access and load observations stored in different formats and sources.

We implemented these proposals in the TerraLib GIS library extension to deal with spatiotemporal data; we developed a new module called "TerraLib ST Module". This module can be downloaded at www.dpi.inpe.br/terralib5. Now, we are working on a plugin for TerraView GIS software. This plugin will provide graphical user interfaces (GUI) to allow users to create the RDF files describing data sources, access spatiotemporal observation sets from these sources, handle these sets and dynamically visualize them in TerraView.

References

Battle, R., Kolas, D. (2012) "Enabling the geospatial semantic web with parliament and GeoSPARQL", Semantic Web Journal, v. 3(4).

Berners-Lee, T., Hendler, J., and Lassila, O. (2001) "The semantic web", Scientific american, v. 284(5), p. 28–37.

Camara, G., Vinhas, L., Queiroz, G. R., Ferreira, K. R., Monteiro, A. M. V., Carvalho, M. T. M., Casanova, M. A. (2008) "TerraLib: An open-source GIS library for large-scale environmental and socio-economic applications", Open Source Approaches to Spatial Data Handling. Berlin, Springer.

ESRI (2010) "Tracking Analyst Tutorial", Available at: http://help.arcgis.com/en/arcgisdesktop/10.0/pdf/tracking-analyst-tutorial.pdf

Ferreira, K. R., Almeida, D. B. F. C., Monteiro, A. M. V. (2014) “A RDF Vocabulary for Spatiotemporal Observation Data Sources”, In: Proceedings of XV Brazilian Symposium on Geoinformatics (GeoInfo), Campos do Jordão, Brazil.

Ferreira, K. R., Camara, G., Monteiro, A. M. V. (2014) “An algebra for spatiotemporal data: From observations to events”, Transactions in GIS, v. 18(2), p. 253–269.

Galton, A. (2004) “Fields and objects in space, time, and space-time”, Spatial Cognition and Computation, v. 1, p. 39—68.

Galton, A., Mizoguchi, R. (2009) “The water falls but the waterfall does not fall: New perspectives on objects, processes and events”, Applied Ontology, v. 4(2), p. 71–107

Gebbert, S., Pebesma, E. (2014) “A temporal GIS for field based environmental modeling”, Environmental Modelling & Software, v. 53, p. 1–12.

Güting, R. H., Schneider, M. (2005) “Moving objects databases”, San Francisco, CA: Morgan Kaufmann.

Hornsby, K., Egenhofer, M. (2000) “Identity-based change: A foundation for spatiotemporal knowledge representation”, International Journal of Geographical Information Science, v. 14, n. 3, p. 207–224.

International Standard Organization (ISO) (2002) “ISO 19108:2002: Geographic information - Temporal schema”, Geneva, Switzerland.

International Standard Organization (ISO) (2004) “ISO 8601:2004: Data elements and interchange formats - Representation of dates and times”, Geneva, Switzerland.

International Standard Organization (ISO) (2008) “ISO 19141:2008 - Geographic information - Schema for moving features”, Geneva, Switzerland.

Kuhn, W. (2009) “A functional ontology of observation and measurement”, In: International Conference on GeoSpatial Semantics. Berlin, Springer LNCS, 5892.

Liu, Y., Goodchild, M. F., Guo, Q., Tian, Y., Wu, L. (2008) “Towards a general field model and its order in GIS”, International Journal of Geographical Information Science, v. 22, n. 6, p. 623–643.

Open Geospatial Consortium (OGC) (2006) “OpenGIS abstract specification topic 6: Schema for coverage geometry and functions”.

Open Geospatial Consortium (OGC) (2010) “Geographic Information: Observations and Measurements - OGC Abstract Specification Topic 20”.

Pelekis, N., Theodoulidis, B., Kopanakis, I., Theodoridis, Y. (2004) “Literature review of spatio-temporal database models” The Knowledge Engineering Review, v. 19

Peuquet, D. J., Hardisty, F. (2010) “STempo: an interactive visualization and statistical environment for discovery and analysis of space-time patterns”, Available at: http://www.geovista.psu.edu/stempo/

Regis, L., Souza, W.V., Furtado, A.F., Fonseca, C.D., Silveira, J.C., Ribeiro, P.J., Melo-Santos, M.A.V., Carvalho, M.S., Monteiro, A.M. (2009) “An Entomological surveillance system based on open spatial Information for participative Dengue control”, Anais da Academia Brasileira de Ciências, v. 81.

Rigaux, P., Scholl, M., Voisard, A.: Spatial Databases with Application to GIS. Morgan Kaufman, San Francisco, (2002).

Veloso, B., Iabrudi, A., Correa T. (2013) “Towards efficient prospective detection of multiple spatio-temporal clusters”, In: Proceedings of XIV Brazilian Symposium on Geoinformatics (GeoInfo), Campos do Jordão, Brazil.

Worboys, M. F. (1994) “A Unified Model for Spatial and Temporal Information”, The Computer Journal, v. 37.

Worboys, M. (2005) “Event-oriented approaches to geographic phenomena”, International Journal of Geographical Information Science, v. 19, p. 1–28.

Yuan, M. (2009) “Challenges and Critical Issues for Temporal GIS Research and Technologies”, In: Handbook of Research on Geoinformatics, IGI Global, Hershey, p. 144–153.