A standards-based framework to foster geospatial data and process interoperability

Gilberto Zonta Pastorello Jr **Rodrigo Dias Arruda Senra** **Claudia Bauzer Medeiros**

Institute of Computing
University of Campinas – UNICAMP
13084-971, Campinas, SP, Brazil
gilbertozp@acm.org rsenra@acm.org cmbm@ic.unicamp.br

Abstract

The quest for interoperability is one of the main driving forces behind international organizations such as OGC and W3C. In parallel, a trend in systems design and development is to break down GIS functionalities into modules that can be composed in an ad hoc manner. This component-driven approach increases flexibility and extensibility. For scientists whose research involves geospatial analysis, however, such initiatives mean more than interoperability and flexibility. These efforts are progressively shielding these users from having to deal with problems such as data representation formats, communication protocols or pre-processing algorithms. Once scientists are allowed to abstract from lower level concerns, they can shift their focus to the design and implementation of the computational models they are interested in. This paper analyzes how interoperability and componentization efforts have this underestimated impact on the design and development perspective. This discussion is illustrated by the description of the design and implementation of WebMAPS, a geospatial information system to support agricultural planning and monitoring. By taking advantage of new results in the above areas, the experience with WebMAPS presents a road map to leverage system design and development by the seamless composition of distributed data sources and processing solutions.

Keywords: geospatial data, geospatial processing, geospatial interoperability, data publication, process publication

**This paper is an extended version of a poster paper [44] presented at ACM GIS 2008, and was supported by FAPESP (grant number 2004/14052-3) and partially financed by CNPq and Microsoft Research-FAPESP Institute.*

1. INTRODUCTION

In geographic information science, interoperability is a key issue, given the wide diversity of available geospatial data and scientific data processing tools. There are many research initiatives to meet this challenge, from data interchange standards and service-oriented architectures (SOA) to user interface design. This paper concentrates on two kinds of interoperability aspects: processes and data [19, 36, 38].

We show that efforts towards these directions have a desirable side effect: they are progressively shielding end users (the scientists) from having to deal with low level data management issues. Indeed, because of the variety of data available, from distinct providers, these scientists are forced to concern themselves with low level implementation details. Interoperability solutions are helping to decrease this problem, thus contributing to bridge the semantic and operational gap between data providers and scientists whose main interest is to design and test computational models that use geospatial data and processes. In this text, the term *model* refers to a computational model representing relevant aspects from natural or artificial phenomena and/or processes that are somehow spatially referenced – e.g. a hurricane (natural phenomenon), or urban traffic (artificial). Models are ultimately embedded in applications – e.g., that provide tools to run or tune models.

The term *user* denotes two categories of people: end-users (i.e., those that will interact with an application, in particular scientists), and designers/developers of models and applications (i.e., those that benefit from the interoperability advantages offered by our framework). When necessary, the text differentiates among them, by using the terms *end-user* and *developer*. Most of the paper concerns developers, even though in the geospatial domain end-users are most often than not required to play the role of developers, being constantly burdened with low-level details. Both kinds of users are positively affected by the framework.

Process interoperability is related to how two (or more) heterogeneous systems can interact. To that end, the systems must have means of determining which operations can/ should be invoked from each other's interface to execute a task. Data interoperability concerns data representation formats and manipulation. To achieve data interoperability, data consumers must be able to interpret each data set according to the same set of concepts. Data and process interoperability are usually treated apart. This unnecessarily complicates application design and development – in fact, these issues are intimately related, since processes consume and produce data.

We categorize approaches to deal with interoperability issues as: standards based, ontologies based and services based. Standards concern reaching an agreement on a domain and specifying interfaces, protocols and data representation formats. Ontologies are also related with a domain, but an *a priori* agreement is not always a requirement. Services present a generic way of encapsulating operations from a system, making them available in a uniform way. Ontologies are out of this paper's scope, being tackled elsewhere [15, 16, 42].

This paper discusses recent efforts in interoperability for geospatial processes and data that are based on standards and services. Our discussion is intertwined with the impacts of such interoperability solutions on the design and development of geospatial applications. Monolithic systems are giving place to component-based distributed architectures [50, 52]. While the former forced scientists to adapt a problem (and their solution to it) to be compatible with a system and its data formats, the latter fosters the adoption of systems and data [28] that will fit the requirements of a new problem. Sensor data applications such as environmental and urban planning [14] are pushing the need for these kinds of solution even harder. They have to rely not only on local, well known, data providers but often on distributed, discovered on-the-fly, heterogeneous systems. In order to leverage design and construction of geospatial applications, however, data must undergo complex sequence of transformations, from providers to consumers. To support the required transformations, the designers of geospatial systems are faced with a multitude of process and data interoperability solutions, which they must somehow choose and compose.

The paper presents two main contributions towards helping solve this problem, namely:

- a conceptual framework that structures those transformation steps into several layers, with clear cut interfaces and responsibilities. Each of these layers addresses one kind of interoperability problem (for instance, access mechanisms, data cleaning, data formatting). This separation helps systems designers to focus on one issue at a time, leading to modular and composable systems. It also shields scientists from having to deal with low-level implementation issues; and,
- a real case study of this framework showing its advantages on data and process interoperability. This case study concerns one of the projects that is using the framework – WebMAPS, a multidisciplinary project involving research in computer science and agricultural and environmental sciences.

Moreover, the paper attacks data and process interoperability problems within a single architecture. As will be seen, this eliminates several obstacles faced by developers, who are often induced to treat data and process interoperability within distinct perspectives.

The rest of the paper is organized as follows. Section 2 proposes a framework for publication of geospatial data and processes, to support application development. Section 3 describes the WebMAPS project and how it is being built under the framework. Section 4 discusses approaches for interoperability, how they fit in the framework and in WebMAPS development. Section 5 presents related work. Section 6 concludes the paper and discusses ongoing work.

2. A FRAMEWORK FOR GEOSPATIAL DATA MANAGEMENT

The architecture of interoperable data management systems is often specified following a basic three-layer cycle: providers (data layer), transformers (service layer) and consumers (client layer). An example is the infrastructure provided by INSPIRE [25], an initiative for the creation of a spatial infrastructure for Europe, with a distributed network of databases, linked by common standards and protocols to ensure compatibility and interoperability of data and services. INSPIRE's architecture (client, services and data layers) includes four main groups of components: user applications, geo-processing and catalog services, catalogs and data repositories. The organization in these three layers is not unique to GIS (Geographic Information Systems) – e.g., see [27] for data warehouse management. Though useful to understand the functionalities provided, this kind of organization is insufficient for designers of geospatial computer models to choose and compose process and data interoperability solutions.

2.1. Framework overview

In order to meet this challenge, we propose an extended framework which induces a methodology for geospatial data management, and the design and implementation of computational models in GIS. This framework, shown in Figure 1, describes a data management cycle for GIS applications – from data acquisition (at the bottom) to data publication (at the top), to be consumed by applications that embed models. This cycle can be repeatedly pipelined: the data publishers of one cycle can become the data providers of the next cycle. As will be seen, the first four layers can be compared to a *Extract-Transform-Load* (ETL) process [2, 20, 22, 48] in data warehouse environments. This organizational cycle provides the basic structure through which process interoperability problems at several levels can be dealt with.

Our full data management cycle has seven layers, which alternate between representing either data or processes. Layers 2, 4, and 6 represent data and boxes with gears (Layers 1, 3, 5, and 7) represent data manipulation operations. The flow is from bottom to top, with the operations being applied to the data on their way up. We point out that not all stages of the cycle are mandatory – e.g., a given intermediate stage may not be needed, or applications may retrieve raw data directly from providers. Furthermore, an entire cycle may be under the control of a single organization (e.g., our case study of Section 3), or distributed on the Web.

The bottom level houses *Data Providers* of many kinds. In the geospatial domain, data providers include sets of files, databases, sensors and data services. Sensors can range from ground-based networks to large satellite embarked multi-spectral electromagnetic sensors.

The upper level houses the applications that embed the *Computational Models*. It is here that end-users are able to interact with all the infrastructure on the layers below. Applications embed model execution, hence allowing scientists to visualize results, and to tune and interact with these models.

Sensor-produced data pose several new challenges to geographic applications. These data have a variety of characteristics that impact how they are stored, processed, published and accessed [23, 53]. Besides the spatio-temporal variability inherent to geospatial data, we single out the following issues particular to sensor data: (i) regularity: production of data in independent blocks or as continuous streams; (ii) transmission: manual readings, wired/wireless transmissions, error introduction, and others; (iii) position: impacts of the sensor relative position on the readings with respect to the observed phenomena; (iv) mobility: relation between sensor movement and its readings. Many other characteristics can be considered, according to the sensor capabilities and the application requirements. The broader the coverage of these aspects, the larger the number of consumers to which the data may be adequate.

Sensor data must be combined with data coming from data services. The latter deliver products provided by organizations that create or enrich a given data set and make it available. These data also have inherent characteristics that influence subsequent manipulation. Examples include issues such as which models were used to produce the data, which parameters were applied to calibrate the model, or how reliable were the data.

Next, we detail what each of the layers encompasses. Layers 1, 2 and 4 can be respectively compared to *extract*, *transform* and *load* phases of an ETL process.

2.2. Layer characterization

Layer 1 (*Acquisition*) hosts data acquisition software, which works as a wrapper to data providers. Machine processable knowledge representation is an important issue in enabling software in this layer to access data with complex characterization from multiple data sources. Standards play an important role to deal with knowledge representation, and are further explored in Section 4, where we discuss how they "wrap" the data management cycles. This layer expresses the *extract* phase of an ETL process.

Layer 2 consists in *Unprocessed data*, obtained directly from data providers in a variety of formats. To be useful within a data management cycle, the data must be adapted to some representation using the characteristics and formats chosen for the cycle. This task is performed in Layer 3. Usually, unprocessed data is stored when there is a need for comparing pre-storage processing solutions, for maintaining historical raw data, or for auditing purposes.

Layer 3 (*Pre-Storage Processing*) represents the processing phase where data is transformed before its storage. Examples include signal processing functions for improving precision, data cleaning for detecting variations or errors, computing statistical parameters to detect acquisition problems, converting temporal and/or spatial resolutions, and testing data format conversions to determine accuracy. This layer corresponds to the *transform* phase of an ETL process.

Layer 4 (*Data Repositories*) corresponds to the storage facility, often a data repository of some kind, such as a database system. Two of the major issues to be dealt with in this layer are problems on what to store and how to fill in the gaps left by several types of acquisition errors. Selecting what is going to be stored is important since the amount of data acquired may be far too large to be stored in full [13, 18]. Given that geospatial systems must also cope with streamed data, this raises the additional issue of providing query mechanisms that can cope with both stored and streamed data [3]. For streamed data, the storage layer may have its role filled by a proxy service with

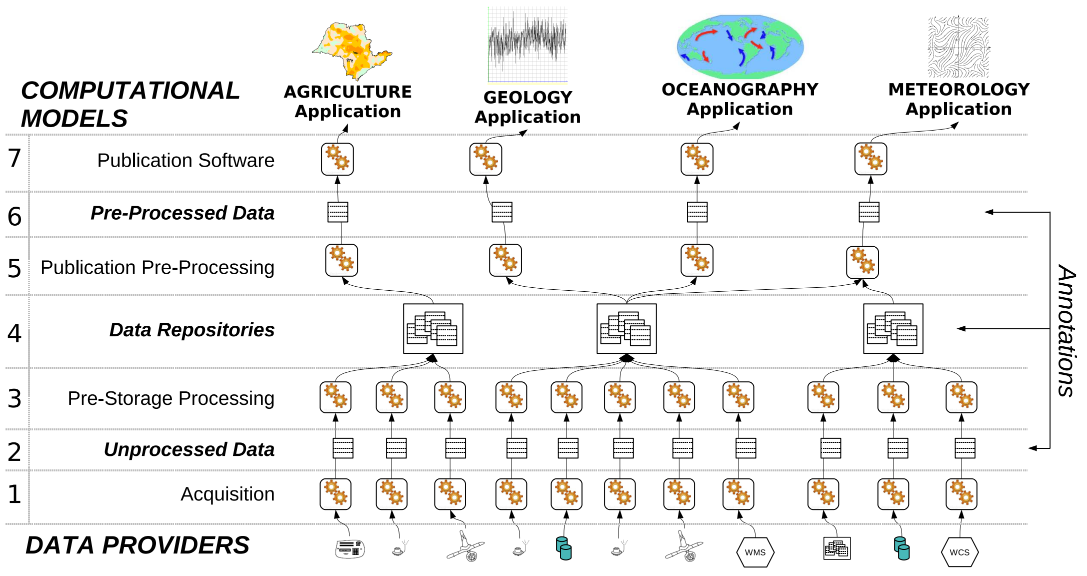

Figure 1. Geospatial data usage scenario.

some kind of caching mechanism. This is more natural for many kinds of applications (e.g., real time traffic monitoring). Queries may need to combine data from several data sources, even both streamed and stored data, possibly using different pre-storage processing operations. This can only be treated adequately in layers that have access to these resources, which is the case for Layer 4. The storage part of this layer is directly related to the *load* phase of an ETL process.

Layer 5 (*Publication Pre-Processing*) is responsible for transforming the data, filtering or augmenting it, in order to meet application requirements. Examples of such requirements include adjusting spatio-temporal resolution, access periodicity and specific presentation formats. Instances of operations to fulfill these requirements include composition operations (e.g., fusion of data from different data sources), scaling operations (e.g., changing temporal or spatial resolution), customization operations or more complex operation compositions. However, as most of these data are georeferenced, the more traditional GIS operations, e.g., see [46], are the most common in this phase. The execution of operations in Layer 5 are guided by application needs while operations executed in Layer 3 are oriented towards storage requirements. Thus, unless the operation was executed in Layer 3 and the result is already available in the repositories, a request from an application is executed in Layer 5.

Layer 6 (*Pre-Processed Data*) contains the pre-processed data sets, ready to be published and consumed by models. The main concern in this layer is data representation, e.g., data format, spatio-temporal resolution, and associated descriptions. An application request specifies the format, with translations applied as needed. Resolution adaptation may require interpolation algorithms for larger resolutions and summarization algorithms for smaller resolutions.

Layers 5 and 6 are not needed in non-shared data scenarios. Their appearance reveals an interesting issue: as models and algorithms become more stable and accepted within a community, they become basic and are taken for granted. This pushes the results of such models and algorithms down from the model layer to layer 5, with impact on interoperability and cooperative scientific work. An example of such migration is the georeferencing of satellite images: in the past, it was a necessary step to perform within geospatial data applications; presently it is available as a default attribute of most published images.

Layer 7 (*Publication Software*) represents the software that will make interfaces to operations and data access mechanisms available to applications. This is achieved by agreement between software providers and application developers. The need for such agreements restricts interoperability among new resources and systems. As an alternative to consensual specification, the software in this layer should provide descriptions that are sufficiently rich to allow applications to determine the suitability of software and data. Different approaches are used by the publication software depending on the requirements of the client applications, e.g., protocols with less overhead for large data sets, or richer protocols for initial stages of communication.

The publication software must also allow applications to select pre-processing operations, among the ones available, to be applied on the data before transmission. The operations are actually provided by lower layers, mainly by layer 5 (publication pre-processing) but a list of them and a selection mechanism must be present on the publication layer. Since the requirements from the applications vary, many different transformation operations may be required before the data can be used. Actually, applications can use either already pre-processed data sets (e.g. from Layer 6) or invoke operations to generate new data sets (e.g., from Layer 5). It is often undesirable or unfeasible to perform these operations within the application [53].

Annotation mechanisms are orthogonal to all layers, using metadata standards or free annotations. In other words, each data layer may be associated with annotation mechanisms that provide semantics to raw, stored or preprocessed data. Metadata have a predefined structure and expected ranges. This allows, for instance, indexing and retrieval based on the metadata, imposing, however, rigid limits. Annotations, on the other hand, have no structure and do not allow indexing, presenting a challenge for retrieval, often requiring content-based techniques [32]. Nevertheless, they allow very flexible descriptions. The proposal in [42] shows how to provide some structure to annotations without hampering flexibility by using references to ontologies. The Section 4 discusses how to improve metadata semantics with standards.

2.3. Remarks on the framework

We point out that making these seven layers explicit is one of the keys to understand and solve the gap between resource providers and systems designers. Our decomposition of geospatial data flow and processing into identifiable layers with clear interfaces and responsibilities leverages application development. Combining solutions from previous layers enables a module on a higher layer (or even an application outside the scope of the layers) to deal with less interoperability issues. The most important aspect, however, is that our organization helps maintainability, reuse and extensibility, allowing developers to include new features at appropriate levels. This is achieved by solving a distinct interoperability issue in each layer and combining the solutions across the layers. To illustrate this, consider the problem of gathering data from two data sources, each using a different combination of access mechanism (e.g., an FTP server and a Web service) and data format (e.g., XML, CSV and binary). Once a module for each of the access mechanisms is available at the bottom layer, all modules for handling different data formats will be able to use one uniform interface to access the data from both providers. This scheme is the same throughout the layers, each layer adds a solution to one interoperability issue. This solution can reuse modules provided to solve issues in lower-level layers – which in fact offer a uniform interface (or data format) to the next layer. Taking these stages into account helps solving several of the interoperability problems raised by the use of distributed geospatial data sources or by the invocation of external services. This will be illustrated next.

3. Putting the framework to use

This section discusses how the proposed framework reflects in the implementation efforts within the WebMAPS project. The framework is also being used in other ongoing projects at the Laboratory of Information Systems (LIS) at the Institute of Computing, University of Campinas. We have chosen to concentrate our discussion on the WebMAPS project, because it is the earliest of such projects, and because it provides enough material to illustrate the advantages of our approach. One product from WebMAPS, vegetation index graphs, is described in Section 3.2 according to the layers presented in Section 2. Two other products are also detailed: one for automation of data acquisition (Paparazzi, see Section 3.3) and another for flexible data publication (see Section 3.4).

3.1. Overview of the WebMAPS project

WebMAPS is a project whose goal is to provide a platform based on Web Services to formulate, perform and evaluate policies and activities in agro-environmental planning.

The project caters to two kinds of users – farmers, and domain experts, such as agronomers or earth scientists. Farmers can enter data on their properties (e.g., production, parcels, crops). They are able to correlate data on their farms to geospatial content available on WebMAPS repositories – e.g., satellite image series or regional boundaries. Experts may want to investigate distinct kinds of data correlation and propose models to explain, monitor, or forecast crop behavior. See some of the tools at `http://www.lis.ic.unicamp.br/projects/webmaps`.

Similar to INSPIRE [25], WebMAPS can also be described using a 3-layer architecture, part of which already implemented. The Client Layer is responsible for user requests, forwarding them to be processed by the middle (Service) layer. The latter contains distinct kinds of modules, such as query processing, workflow specification, and ontology management. The Data Layer contains WebMAPS data repositories, including primary raw data (e.g., product classification from Brazilian official sources) and derived data (e.g., composite images). Geospatial data sets include satellite images, and coordinates of county boundaries. Additional data sources include information on properties, agricultural products and so on. Data are stored in the PostGreSQL database management system, with PostGIS extension. At present, most of the services in WebMAPS are being implemented as software modules, for rapid prototyping and testing by end-users.

This kind of 3-tier architecture is useful for a high level description of the system's functionalities. However, as stressed in Section 2, it is not adequate from an interoperability perspective. The sections that follow discuss how we use our 7-layer framework of Section 2 to specify and develop some of the products offered by WebMAPS. In particular, we discuss three kinds of products: (i) the dynamic computation of NDVI graphs, starting from the acquisition of satellite images; (ii) a tool for automated image acquisition; and (iii) the interoperation with Google Maps. The first item is an example that spreads throughout most of the layers, while the last two focus on the bottom and top of the framework, respectively.

3.2. Illustrating the data management framework in WebMAPS

In this section, we describe one of the devised WebMAPS' products that is partially implemented and adheres to the layering described in section 2: computing historical NDVI profiles for a given region and period.

NDVI (*Normalized Difference Vegetation Index*) is a vegetation index. It is correlated to biomass conditions of vegetation and is widely used in distinct kinds of contexts – e.g. agriculture, biodiversity. An NDVI graph plots the average NDVI pixel value in a region through time from a temporal series of images. This can be used for crop monitoring and prediction [5, 45]. For example, in the sugar cane culture, a curve with higher values may indicate a product with better quality.

One of the functionalities available from WebMAPS is the construction of such graphs. The user selects a region of interest R and a period T and the system computes the graph from a temporal series of satellite images, plotting the NDVI evolution for that region and period, as depicted in Figure 3.

NDVI graphs require two kinds of data – those acquired periodically (satellite images) and those that, once acquired, are only sporadically updated (e.g., county boundaries). This section describes the management cycle for these data within WebMAPS. We will not enter into details of acquisition periodicity nor procedures to refresh data, but such issues are embedded into constraints treated by our 7-layer framework. Figure 2 shows the main phases of the workflow that specifies the computation of the graph, following the layers of Figure 1. This workflow is shown at its more abstract level, but each step can encapsulate several processes. Moreover, though not shown in the Figure, it contains loops and cycles, which are not relevant for understanding our case study. We point out that this example does not have issues to be dealt with in layers 2 and 6. Applications that, for instance, are heavily dependent on data representation formats, data encoding or associated descriptions (metadata) would need specific solutions in these layers.

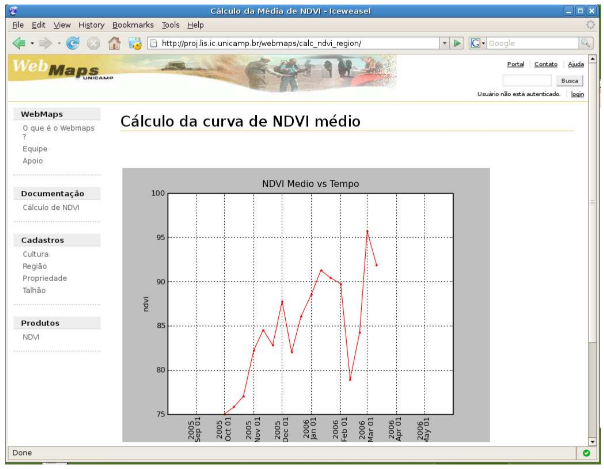

Figure 3. Example of an average NDVI curve for Campinas region.

3.2.1. Data Acquisition: There are many satellite imagery providers. For NDVI analysis, WebMAPS' agro-scientists have chosen to use pre- computed NDVI images provided by NASA from MODIS sensors [37, 51]. Here we faced typical problems of geospatial data acquisition. Each image depicts a geographical region much larger than the ones for which this first version of WebMAPS is being conceived (Brazil's southeast). Moreover, retrieving each image meant browsing the NASA web site to find the download link for that image. Thus, assembling our image database became a laborious, tedious and time-consuming task. To improve on that, we have developed Paparazzi, a tool to automate the retrieval of remote data sets – see Section 3.3.

The second data type needed are vector-based coordinates, corresponding to the geographical regions of interest. WebMAPS offers two options: (i) ad-hoc manual definition of the region, described in *Well-Known Text Format* (a.k.a WKT) [10], a standard from the *Open Geospatial Consortium* (OGC); or, (ii) importing geospatial vector shapefiles. Brazilian county geometry shapefiles were imported from IBGE (Brazilian National Geographic Institute).

The last data type we need are textual descriptions of crops and their attributes. Here we applied screen scraping [12, 21] techniques to fetch produce code, popular name, scientific name and description from the Brazilian Ministry of Agriculture Web Portal. See Section 4 for details on these techniques.

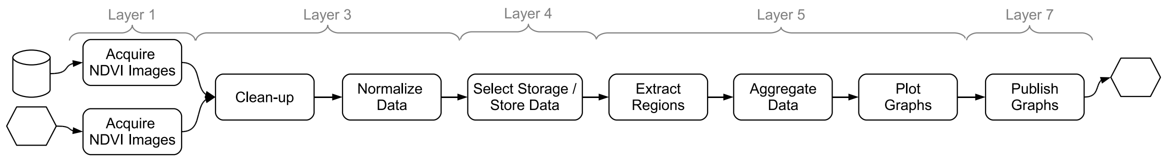

Figure 2. Computation and publication of NDVI series for a region.

3.2.2. Unprocessed Data: Satellite images retrieved from NASA using Paparazzi and shapefiles retrieved from IBGE are encapsulated in temporary files, for subsequent quality checking. The rest of the data used goes directly to Layer 3 (Pre-Storage Processing). Here, we can already see the advantages of our multi-layer framework, which allows determining which stages should be followed for each kind of data.

3.2.3. Pre-Storage Processing: In possession of unprocessed data, we proceed to the pre-storage processing phase. There are three main concerns here: corruption detection, data normalization and assembly of the data sets. These concerns are not always present (e.g., if the data provided already have such issues solved). In particular, our NDVI graph construction example does not need to deal with the assembly of data sets.

Corruption detection is mandatory and is made explicit in our framework. First, data providers are never 100% reliable, and the acquired data may be already corrupted in its provider's domain. Second, data corruption can occur during the acquisition phase. Here, the encapsulated unprocessed files containing satellite data and geometries have their integrity automatically checked (e.g., by checksum algorithms). Corrupted or partially retrieved files are removed.

The third (textual) record type is more challenging, because information is less structured, the domain of values is open and not fully-known and we lack fail-proof tools to verify corruption. For our textual data resources (based on official government crop classification) we performed a manual check. Additional procedures are left to future work.

Data normalization is a recommended step to make data processing easier and more efficient. We automatically convert all files to a single and uniform representation format, and all measurement units to the same system. Thus, we have chosen to (a) store satellite imagery into GeoTIFF [47] files, converting into this format whenever needed, (b) convert shapefiles and textual geometries (WKT) into *Well-known Binary* representations (WKB) stored into PostGIS, and (c) represent all geographical coordinates to latitude/ longitude according to WGS84 reference ellipsoid. WKB [10] is also a standard from OGC.

Data set assembly is the last pre-storage processing step we need, and consists in putting together coherent spatio-temporal units – e.g., in our example, creating a composite NDVI image from a mosaic of acquired NDVI images. This includes issues which are sometimes called detecting the FoU (fitness of use) of a data set – see [30]. Here, the first (temporal) problem occurs when there are gaps in a time series – e.g., due to communication or device failure. The second (spatial) problem occurs when missing spatial data create "holes" in a data set (e.g., when an image mosaic has missing parts). Both problems can occur at the same time. Spatio-temporal gap problems are very common when using data from sensor networks – e.g., sensors may stop providing data for a period of time, causing problems in analyses.

There are three basic approaches to assembly problems: (i) acquire new data to fill the spatial and/or temporal gaps; (ii) apply interpolation, probabilistic or inference procedures to fill the gaps; and, (iii) mark the gaps, and forward the solution to some other layer (e.g., query processing will have to take the gaps into consideration). For satellite imagery, we have implemented the first approach, using data fetching retries and fallback data providers. For rainfall time series, we use the second approach. We have developed algorithms in which missing values are filled by combining spatial interpolation with historical data [33]. These algorithms are used by the Brazilian Ministry of Agriculture to maintain its rainfall series database for the whole country.

3.2.4. Data Repositories: Once the data are preprocessed, they are ready for storage. We use two types of storage: a relational database and the filesystem. Crop descriptions, geometries, textual properties, and data set descriptions are stored in PostgreSQL/PostGIS. Raster images in GeoTIFF format are stored in JFS (or XFS) filesystem partitions. We have chosen these partition types because they present good performance for large files [6]. So far, we have not used streamed sensor data. Our preliminary experiments with these data appear in [43]. Streamed sensor data for agricultural purposes are being handled within our eFarms project (`http://www.lis.ic.unicamp.br/projects/efarms`), which will act as a sensor data provider to WebMAPS.

3.2.5. Publication Pre-processing: This phase concerns transforming information, and ultimately preparing it for user consumption. In our example, this means (i) computing the average NDVI pixel value for the region R defined by the user; and, (ii) iterating (i) for the input time span T. These steps are performed automatically without user intervention.

The images of interest are already stored in the repositories. Steps (i) and (ii) consist in building a bitmap mask for R and applying this (cached) mask to all NDVI images within the desired time frame T, extracting the region (pixels) of interest and computing their average. The graph is constructed for these average values.

3.2.6. Data Publication: Data publication is the last phase in the processing workflow. In our case study, the NDVI graphs constructed in the previous phase are published as images, embedded or not in HTML pages. Other formats could also be adopted here: text files with pairs <georeferenced point, value>, tables with the values for regions, etc.

3.3. Automating Acquisition: Paparazzi

Paparazzi is a command-line tool we developed to automate the acquisition of satellite imagery by means of screen scraping techniques [24, 29, 31]. Paparazzi is a specialized web crawler, hand-crafted to fetch data from specific target web sites. Paparazzi is an example of a tool that was implemented within the scope of Layer 1. Paparazzi is worth using whenever the number of files to be retrieved is large, and hyperlinks to target files are not concentrated in a single page, but scattered across several pages, as is the case with NASA MODIS images. Consider the following Paparazzi command line:

```
paparazzi.py -b 2008-01-01 -e 2008-05-31 -s Brazil4 -p
250m -m 2 -r
```

It is a request for all images (`-r`) from January 1, 2008 (`-b`) to May 31, 2008 (`-e`), for the geographical region named *Brazil4* (`-s`). Each image retrieved should have spatial resolution (`-p`) of 250 meters per pixel and represent NDVI measures (`-m`). *Brazil4* is a specific name used by NASA [37] to designate a given area in South America that covers SE Brasil.

This same task required downloading 152 images corresponding roughly to 7 Gb in size. If done manually, for each image, the user needs to visit three different web pages prior to starting a 50 Mb file download. In the first page the user selects the subset (Brazil4). In the next page the user selects data product (NDVI) and image resolution (250m). Finally, in the last page, the user selects the file format and starts the image download. Therefore, precious user time can be saved, if the researcher relinquishes control and responsibility of the iterative acquisition process to Paparazzi. Moreover, when adopted as a software library, Paparazzi acts in reaction to user queries over incomplete data sets, trying to fill-in gaps on demand.

3.4. Flexible publication

WebMAPS innovates allowing data produced in any of the framework phases to be directly accessible in many representations. Therefore, WebMAPS is not just a blackbox automating GIS procedures. It is also a data gateway fostering scientific information sharing and allowing experimental results to be reproduced and validated. This is facilitated by isolating the responsibilities of each framework layer.

In particular, the three data types handled (images, geometries and text) are published by means of standard protocols and representations, explained in Section 4. Satellite imagery data is accessible through an OpenDAP interface; textual data and metadata are available as HTML pages annotated with microformats; and geometries are available as KML views.

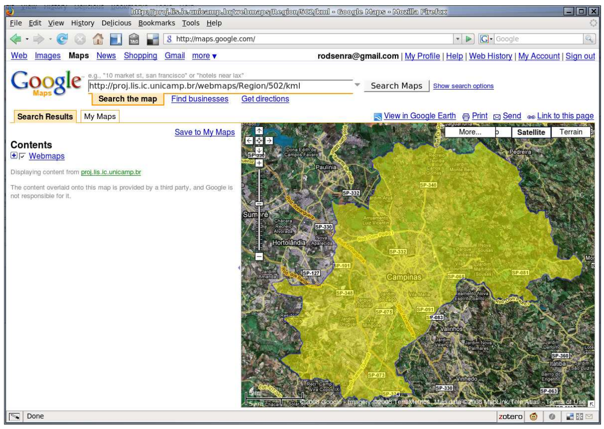

Figure 4. Google accessing data from WebMAPS in KML format.

Thanks to this, WebMAPS can act as a data provider and a data client. We exemplify this by showing its interaction with Google Maps. Figure 4 depicts Google Maps obtaining a geometry resource served in KML format from WebMAPS. Here, WebMAPS acts as a data provider and mediator, re-distributing geometries acquired from an authoritative source (Brazilian Geographic Institute), and transformed into a suitable format to feed Google Maps. In Figure 5, we depict WebMAPS acting as a client of Google's map rendering service. The map rendered by Google Maps (Figure 4) is mashed-up with results from a user query, composing the web page shown in Figure 5. The query results comprehend textual metadata and a NDVI graph for the given region and time frame. For visualization sake, the chart is an overlay, not representing the original page layout.

This interaction pattern between WebMAPS and Google Maps is a combination of resource-oriented (from

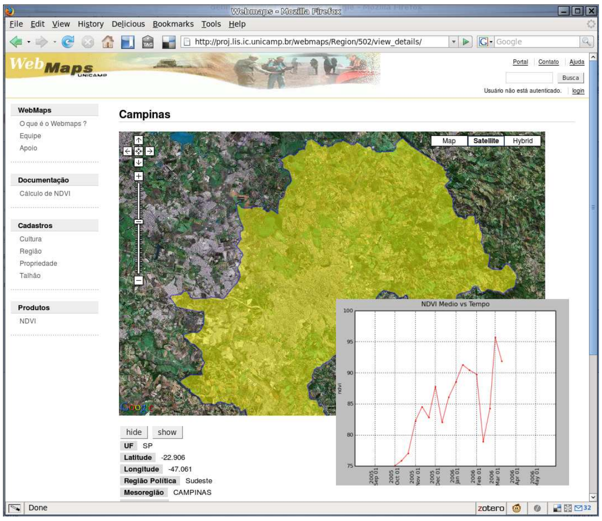

Figure 5. WebMAPS embedding map generated by Google Maps.

WebMAPS) and service-oriented (from Google Maps) paradigms. We further discuss these approaches in Section 4. In this particular example, the use of KML and WKT enabled us to rapidly build a prototype for cartographic visualization, including satellite image overlays provided by Google Maps. End users are rapidly able to visually assess the quality of the data, and test the outcomes of different analyses. Hence, standards offer much more than interoperability. Their use has sped up the validation of user requirements in terms of interaction needs. More importantly, it has leveraged application development, so that users can start testing their ideas much sooner, while we work on other system issues. This does not mean that WebMAPS will necessarily always rely on Google Maps for cartographic rendering and interaction - we are also experimenting with other kinds of Web service-based solutions (see [16] for our use of GeoServer to publish GML data for biodiversity systems).

4. Interoperability Approaches

This section characterizes the standards and services approaches to interoperability and show how they are contemplated within our framework, with examples from WebMAPS. We point out that ontologies are another very important approach to interoperability. In our framework, they intervene at all transversal annotation stages of Figure 2. They nevertheless are outside the scope of this paper.

4.1. Selected Standards

Standards represent an agreement among research groups and are the main focus of many institutions such as OGC or W3C. From the data interoperability perspective, standards deal with representation and formatting issues. OGC's *Geographic Markup Language* GML [39] is an example of such a standard. It is an XML-based specification for geospatial data interchange. Process interoperability specifications can base their input/output formats in GML.

From the process interoperability perspective, standards are used in the specification of protocols, interfaces and descriptions of processes. Examples include OpenDap and OGC standards. OGC's main general-use standards for geospatial process interoperability are the *Web Feature Service* (WFS), the *Web Coverage Service* (WCS) and the *Web Map Service* (WMS) [39]. These standards specify the access mechanisms to, respectively, vector data, raster data and renderized maps. Vector data describe geographic features using their geometry (points, lines, polygons). Raster data represent geographic areas as arrays of cells (e.g., images). The access mechanisms are to be implemented as Web services.

The recent *Web Processing Service* (WPS) [40] specification concerns the publication of geospatial processes, a main concern in this paper. A process may be an algorithm, a calculation or a model that manipulates geospatial data. Although WPS does not describe the specific behavior of an operation, it provides general description mechanisms, such as *Profiles* and *ProcessDescriptions* [40] and support for data encoded in GML. This, however, still leaves room for semantic mismatches.

Standards must be present at least in the frontiers of our data manipulation cycle, "wrapping" it (see Section 2). The communication interfaces for data acquisition and publication are the two points where these solutions are most useful: as seen in Section 3.4, WebMAPS can be seen as a client application and a data provider to client applications. As a server, WebMAPS strives to adhere to standards, to enable interoperation with other systems. As a client, taking advantage of standard interfaces is important, but, as will be seen, being able to handle involuntary, non-standardized, access mechanisms might be equally important. As part of those efforts, its development is adopting Web services and SOAP protocols, OpenDAP, Microformats and KML, discussed next.

OpenDAP is an acronym for *Open-source Project for a Network Data Access Protocol*. It consists of a data transport architecture and HTTP-based protocol capable of encapsulating structured data, annotating the data with attributes and adding semantics that describe the data. One of its features is the ability to retrieve file subsets, and aggregate data from several files in one transfer operation. It is being increasingly adopted by earth scientists to publish their data – e.g., in oceanography [9, 11, 52]. As exemplified by [52], this allows scientists to exchange and visualize results of complex models. In our framework, OpenDAP can be used as a means to receive and publish data, in layers 1 and 7. For instance, images are acquired and served by WebMAPS using OpenDAP. In the first case, it is at the receiving end (Layers 1 and 2), while in the second case it is at the top of the cycle.

Microformat is a web-based data formatting approach to re-use existing content as metadata, through standard annotations conveyed by XHTML (or HTML) classes and attributes. The intention is to allow information targeted to end-users to be also software processable. In other words, the layout and formatting markup are also used to perform semantic annotations. Microformats replace more complicated methods of automated processing, such as natural language processing or screen scraping. Their use has direct impact in the representation of data in Layer 6, after being generated in Layer 5 along with other transformation processes.

In particular, Geo is a microformat used for marking up WGS84 geographical coordinates (lat,long) in XHTML. Figure 6 presents an example of the use of this microformat in an XHTML page. This allows parsing tools to mine for pages that contain coordinates in this format. This allows these pages to be rendered using this geospatial information, e.g., in a mapping tool or loading the coordinates into a GPS device.

```
<div class="geo">         Campinas:
  <span class="latitude">  -22.906 </span>;
  <span class="longitude"> -47.061 </span>
</div>
```

Figure 6. Example of the Geo microformat in an XHTML page.

KML [41] is an XML-based language schema for expressing geographic annotation and visualization for 2D and 3D Earth browsers. It was initially developed for use with Google Earth. The KML 2.2 specification was accepted as an OGC standard, ensuring its status as an open standard for all geobrowsers. In WebMAPS, our geometry files are represented in KML and accessed by Google, in which case we are acting as data providers for another data management cycle. Figure 7 shows an example of a KML document where the city of Campinas is represented as a point (its centroid).

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
<Placemark>
  <description>City of Campinas</description>
  <name>Campinas</name>
  <Point>
    <coordinates>-22.906,-47.061</coordinates>
  </Point>
</Placemark>
</kml>
```

Figure 7. Example of a KML document.

The perspective of the data providers on how much effort they put in providing interoperable access mechanisms can be seen as voluntary and/or involuntary. The voluntary point of view is the one we have been discussing so far in this section. It is when the data provider willingly serves its data by means of well-known standardized interfaces and protocols, fostering data interchange between systems. Voluntary access mechanism usually comply to some extent to a standard, in an effort to make the data more easily accessible. The involuntary viewpoint encompasses data providers that are only concerned with data consumption from human users, providing no facilities for external systems interested in obtaining the same information. The reasons vary from lack of resources to the deliberate wish to prevent inter-systems data sharing. Therefore, involuntary access mechanisms usually do not have standardization as a main concern, even though they may comply to standards on occasion.

In order to include involuntary data providers in WebMAPS, we have adopted techniques from the information extraction research field. One of these techniques is called *screen scraping*, in which a computer program extracts data from the displayed output of another program. Search engines and web crawlers use web scraping techniques. Indeed, Web pages are built using text-based mark-up languages, and frequently mix content with presentation. Therefore, web screen scrapers extract machine-friendly data from XHTML and other markup formats.

We believe that automated acquisition is important to bridge the test of computational models against data from several geospatial resource providers. The Paparazzi toolset, discussed in Section 3.3, is a step towards that goal.

4.2. Services Approaches

In the services category of interoperability, there are two paradigms competing in the Web: Service-oriented architectures (SOA) and Resource-oriented architectures (ROA). SOA is a direct evolution of concepts born from distributed computing theory and modular programming practices. It is an architecture where functionality is grouped around processes and packaged as interoperable RPC-style services, loosely coupled with operating systems or programming languages. SOA's goal is to facilitate the composition of distributed web services, through the standardization of interfacing, reliable messaging, transactions and security. SOA's philosophy transcends the Web medium and could be successfully applied to other contexts.

On the other hand, ROA is intimately related to the Web. It rescues the principle of Representational State Transfer (REST), defined in [17]. REST outlines how resources are defined, addressed and accessed through simple interfaces, where domain-specific data is transmitted over HTTP without any additional messaging layer or session tracking mechanisms. ROA design aims for the

Web's scalability, and it is defined by five main principles. First, application state and functionality are divided into resources. Second, a resource is uniquely addressable by means of hypermedia links. Third, all resources share the same constrained, uniform and well-defined interface. Fourth, resources support the HTTP protocol operations: GET, PUT, POST, DELETE, HEAD, OPTIONS. Finally, protocols should ideally be stateless, cacheable, layered and client-server oriented. ROA is more scalable than SOA, and easier to implement due to its uniform interface and adherence to Web model and standards.

SOA and ROA are complementary paradigms; together they maximize interoperability. For this reason, we advocate the adoption of hybrid architectures, such as in WebMAPS. As mentioned in Section 3.4, WebMAPS uses SOA when we act as a client to Google, and ROA when Google plays the role of WebMAPS client.

5. Related Work

To the best of our knowledge, no previous work considers all the phases covered in our framework. However, there are proposals that are related with a few of the layers, some of which are presented here. They cover the entire spectrum of solutions discussed in the paper, from data and process interoperability using standards and services, to user interface aspects.

There are many studies concerning use of standards, usually restricted to just one of our layers. For instance, Aim4GDI [1] uses OGC standards for accessing distributed data sources and creating composite results. It also extracts metadata from these sources, composing them in RDF for later querying (using SPARQL) and publication (in an ontology description language). However, it only considers issues at the data access and interchange level, not covering processing resources and their interoperability.

The work presented in [26] considers the use of standards for both data and process interoperability, for distributed sources. Their solution consists on a framework based on the ISO19100 series of standards. The paper materializes the framework in a travel guide system called MTGS. However, limiting the standards considered for interoperability into a single standards source hampers the construction of multi-disciplinary models and applications, preventing their evolution. This is remarked by [34], which discusses the evolution of the GML standard and the importance of integrating it with standards from other application areas.

Interoperability through services is also common, in particular taking advantage of WFS, WCS and WMS. The work of [11], for instance, describes initiatives towards combining communication and access standards, e.g., providing common grounds for OGC's WFS and WCS to work side by side with OpenDAP to access oceanographic data. Their effort concurs with ours in the sense that combining different standards into systems design is a way of leveraging interoperability. Sensor networks can also be encapsulated according to the class of service provided [8]. In such a case, services are even more appropriate.

Our main concern, however, is to provide adequate support to flexible system development. From this point of view, the motivation of GeoModeler [52] is the closest to ours, making geospatial resources more accessible to applications running models. GeoModeler is a software framework that combines software components from a GIS with modeling and simulation software, ultimately allowing various forms of analysis and visualization of oceanographic data. Its approach, however, deals with construction of centralized systems and software components interoperability in such systems. It does not consider, for instance, data acquisition and publication issues.

Our layers stimulate data and process interoperability. One concern (e.g., Layers 3 and 5) is to ensure data quality. This kind of emphasis is undertaken, for instance, by Thakkar et al. [49]. The paper presents a mediator that considers data quality as the driving force of the integration process. Their integration approach involves comparing and evaluating different data providers and keeping information on this evaluation available alongside with the data. However, they do not address process interoperability issues nor take full advantage of metadata and representation standards.

Finally, there are a variety of proposals that consider user aspects: the use of contextual information [7], or interactions in which the user is a computational system [30], or humans [4].

The influence of contextual information on the semantic descriptions of geospatial data and processes is discussed in [7]. The paper also evaluates how context impacts on the user interaction mechanisms in geospatial system user interfaces. It proposes a framework that takes advantage of contextual information and description representations in ontologies to help guide the user through the composition of distributed data and processes. Although we do not explicitly use contextual information, our goal is similar. Our solution favors the adoption of standards and services to provide this effect, with advantages on precision of terms and disadvantages in flexibility. Our annotations can also provide additional contextual information.

The focus of the framework proposed in [30] is to evaluate the suitability of geospatial time series to the requirements of a given application. Once the suitability is calculated, it can be applied to assess the results produced by the application, helping determine the suitability of such results to be used as input by other applications. In our solution, application requirements are also a major concern, but we consider the impacts of interoperability solutions in meeting such requirements instead of trying to tackle them directly.

Finally, visualization of spatial and non-spatial data on the Web is the main concern of [4]. They argue that access to geospatial data aimed at visualization should be easier and more efficient than transferring whole data sets to be processed locally. They propose a browser that supports queries to geospatial services, invoking remote processes and getting the results incrementally. Their solution goes notably in the direction of leveraging the transition proposed by us (from focus on resources to focus on models), with the limitation of not considering distributed data and processing sources.

6. Concluding Remarks

This paper presented a framework that analyzes the management of geospatial data from a life cycle perspective. This framework is being validated in the design and development several projects within the Laboratory of Information Systems of the Institute of Computing, UNICAMP.

By isolating each layer in the cycle, with clear interfaces and tasks, the framework induces a methodology to design and develop interoperable geographic applications. Whereas related research concentrates on providing standards or services for one given data transformation stage, we show how these efforts can be seamlessly interconnected. This allows users to shift their focus from the technology being used to the models being constructed. Besides implementation efforts for the WebMAPS project, we are also applying the framework to the development of the eFarms project, which is centered on managing data from ground-based sensing devices. The framework not only helped understanding and implementing solutions to the problems in sensor data management, but it also made clearer the possible interactions with other solutions (such as the ones from WebMAPS) and which modules from these solutions could be reused.

Future and ongoing work involve both theoretical and practical issues. We are examining additional access standards to be included in WebMAPS, both from the communication and data representation points of view. Another research issue involves the use of ontology-based techniques to speed up query processing and annotate data and processes. Again, we point out that we have not considered ontologies in this work, even though they are another important means of improving data and process interoperability. For detail on on our work in this direction, the reader is referred to [16]. Ontologies are also subject of ongoing work as part of our annotation efforts [42]. Still yet another ongoing effort is to incorporate our work about diagnosing similarity of oscillation trends in time series [35]. Finally, we are investigating the possibility of storing satellite image files in PostGIS and let them be handled by the Rasdaman system (`http://www.rasdaman.com/`).

Acknowledgements. This work was supported by FAPESP (grant number 2004/14052-3) and partially financed by CNPq (WebMAPS projects) and by the Microsoft Research-FAPESP Institute (eFarms project).

References

[1] T. Aditya and M.-J. Kraak. Aim4GDI: Facilitating the Synthesis of GDI Resources through Mapping and Superimpositions of Metadata Summaries. *Geoinformatica*, 11(4):459–478, 2007.

[2] H. Agrawal, G. Chafle, S. Goyal, S. Mittal, and S. Mukherjea. An Enhanced Extract-Transform-Load System for Migrating Data in Telecom Billing. In *IEEE 24th Int. Conf. on Data Engineering*, pages 1277–1286, 2008.

[3] H. Balakrishnan, M. Balazinska, D. Carney, U. Çetintemel, M. Cherniack, C. Convey, E. F. Galvez, J. Salz, M. Stonebraker, N. Tatbul, R. Tibbetts, and S. B. Zdonik. Retrospective on Aurora. *The VLDB Journal*, 13(4):370–383, 2004.

[4] F. Brabec and H. Samet. Client-Based Spatial Browsing on the World Wide Web. *IEEE Internet Computing*, 11(1):52–59, 2007.

[5] M. E. Brown, J. E. Pinzon, K. Didan, J. T. Morisette, and C. J. Tucker. Evaluation of the consistency of long-term NDVI time series derived from AVHRR, SPOT-vegetation, SeaWiFS, MODIS, and Landsat ETM+ sensors. *IEEE Transactions on Geoscience and Remote Sensing*, 44(7):1787–1793, 2006.

[6] R. Bryant, R. Forester, and J. Hawkes. Filesystem Performance and Scalability in Linux 2.4.17. In *Proc. of the USENIX Annual Technical Conference (FREENIX Track)*, pages 259–274, 2002.

[7] G. Cai. Contextualization of Geospatial Database Semantics for Human—GIS Interaction. *Geoinformatica,* 11(2):217–237, 2007.

[8] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng,

M. Musolesi, K. Fodor, and G.-S. Ahn. The Rise of People-Centric Sensing. *Internet Computing, IEEE*, 12(4):12–21, 2008.

[9] L. H. Chambers, E. J. Alston, D. D. Diones, S. W. Moore, P. C. Oots, and C. S. Phelps. The MY NASA DATA Project: Tools for Knowledge Sharing and Discovery. In *Proc. of the 6th Annual NASA Earth Science Technology Conference*, 2006.

[10] Open Geospatial Consortium. OpenGIS Implementation Specification for Geographic Information – Simple Feature Access. `http://www.opengeospatial.org/standards/sfa` (as of September 2008).

[11] P. Cornillon, J. Caron, T. Burk, and D. Holloway. Data access interoperability within IOOS. In *Proc. of MTS/IEEE OCEANS*, pages 1790–1792 Vol. 2, 2005.

[12] V. Crescenzi and G. Mecca. Automatic Information Extraction from Large Websites. *Journal of the ACM*, 51(5):731–779, 2004.

[13] T. Critchlow, K. Fidelis, M. Ganesh, R. Musick, and T. Slezak. DataFoundry: information management for scientific data. *IEEE Transactions on Information Technology in Biomedicine*, 4(1):52–57, 2000.

[14] D. Cuff, M. Hansen, and J. Kang. Urban sensing: out of the woods. *Communications of the ACM*, 51(3):24–33, 2008.

[15] J. Daltio and C. B. Medeiros. Aondê: An Ontology Web Service for Interoperability across Biodiversity Applications. *Information Systems*, 33(7-8):724–753, 2008.

[16] J. Daltio, C. B. Medeiros, L. Gomes Jr, and T. M. Lewinsohn. A framework to process complex biodiversity queries. In *Proc. of the ACM Symposium on Applied Computing*, pages 2293–2297, 2008.

[17] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.

[18] D. Ganesan, D. Estrin, and J. Heidemann. Dimensions: why do we need a new data handling architecture for sensor networks? *ACM SIGCOMM Computer Communication Review*, 33(1):143–148, 2003.

[19] M. F. Goodchild, M. J. Egenhofer, R. Fegeas, and C. A. Kottman (editors). *Interoperating Geographic Information Systems*. Kluwer, 1999.

[20] L. Haas. Beauty and the beast: The theory and practice of information integration. In *Proc. of the 11th Int. Conf. on Database Theory*, pages 28–43, 2007.

[21] B. Habegger and M. Quafafou. Building Web Information Extraction Tasks. In *Proc. of the IEEE/WIC/ACM Int. Conf. on Web Intelligence*, pages 349–355, 2004.

[22] A. Y. Halevy, N. Ashish, D. Bitton, M. Carey, D. Draper, J. Pollock, A. Rosenthal, and V. Sikka. Enterprise information integration: successes, challenges and controversies. In *Proc. of the 24th ACM SIGMOD Int. Conf. on Management of Data*, pages 778–787, 2005.

[23] J. M. Hellerstein, W. Hong, and S. Madden. The Sensor Spectrum: Technology, Trends and Requirements. *SIGMOD Record*, 32(4):22–27, 2003.

[24] D. Huynh, S. Mazzocchi, and D. Karger. Piggy Bank: Experience the Semantic Web inside your web browser. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(1):16–27, 2007.

[25] INSPIRE. Infrastructure for spatial information in the europe. `http://www.ec-gis.org/inspire/` (as of September 2008).

[26] S.-G. Jang and T. J. Kim. Modeling an Interoperable Multimodal Travel Guide System using the ISO 19100 Series of International Standards. In *Proc. of the 14th Annual ACM Int. Symposium on Advances in Geographic Information Systems*, pages 115–122, 2006.

[27] M. Jarke, M. A. Jeusfeld, C. Quix, and P. Vassiliadis. Architecture and quality in data warehouses: An extended repository approach. *Information Sytems*, 24(3):229–253, 1999.

[28] V. Kantere and T. Sellis. Handling spatial data in distributed environments. In *Proc. 15th Annual ACM Int. Symp. on Advances in Geographic Information Systems*, 2007.

[29] R. Kosala and H. Blockeel. Web mining research: a survey. *ACM SIGKDD Explorations Newsletter*, 2(1):1–15, 2000.

[30] L. Fu and L.-K. Soh and A. Samal. Techniques for Computing Fitness of Use (FoU) for Time Series Datasets with Applications in the Geospatial Domain. *Geoinformatica*, 12(1):91–115, 2008.

[31] A. H. F. Laender, B. A. Ribeiro-Neto, A. S. da Silva, and J. S. Teixeira. A brief survey of web data extraction tools. *SIGMOD Record*, 31(2):84–93, 2002.

[32] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2(1):1–19, 2006.

[33] J. G. S. Lima, C. B. Medeiros, and E. D. Assad. Integration of heterogeneous pluviometric data for crop forecasts. In *Proc. of the 5th Brazilian Symposium on GeoInformatics*, 2003.

[34] C.-T. Lu, R. F. Santos Jr, L. N. Sripada, and Y. Kou. Advances in GML for Geospatial Applications. *Geoinformatica*, 11(1):131–157, 2007.

[35] L. E. Mariote, C. B. Medeiros, and I. Lee. Diagnosing Similarity of Oscillation Trends in Time Series. In *Proc. of the 7th IEEE Int. Conf. on Data Mining Workshops*, pages 643–648, 2007.

[36] C. B. Medeiros and A. C. Alencar. Data quality and interoperability in GIS (In Potuguese). In *Proc. of the 1st Brazilian Symposium on GeoInformatics*, 1999.

[37] NASA. MODIS Rapid Response System. `http://rapidfire.sci.gsfc.nasa.gov/` (as of September 2008).

[38] NSF/NCGIA/OpenGIS, editor. *Int. Conf. and Workshop on Interoperating Geographic Information Systems*, 1997. `http://www.ncgia.ucsb.edu/conf/interop97` (as of June 2008).

[39] OGC. OpenGIS Reference Model (ORM). `http://portal.opengeospatial.org/files/?artifact_id=3836` (as of September 2008).

[40] OGC. OpenGIS Web Processing Service. `http://www.opengeospatial.org/standards/wps` (as of September 2008).

[41] Open Geospatial Consortium, Inc. KML Implementation Standard. `http://www.opengeospatial.org/standards/kml/` (as of September 2008).

[42] G. Z. Pastorello Jr, J. Daltio, and C. B. Medeiros. Multimedia Semantic Annotation Propagation. In *Proc. 1st IEEE Int. Workshop on Data Semantics for Multimedia Systems and Applications (DSMSA) – 10th IEEE Int. Symposium on Multimedia (ISM)*, pages 509–514, 2008.

[43] G. Z. Pastorello Jr, C. B. Medeiros, and A. Santanchè. Accessing and Processing Sensing Data. In *Proc. 11th IEEE Int. Conf. on Computational Science and Engineering*, pages 353–360, 2008.

[44] G. Z. Pastorello Jr, R. D. A. Senra, and C. B. Medeiros. Bridging the gap between geospatial resource providers and model developers. In *Proc. 16th ACM Int. Conf. on Advances in Geographic Information Systems*, pages 379–382, 2008.

[45] P.J. Pinter Jr, J.L. Hatfield, J.S. Schepers, E.M. Barnes, M.S. Moran, C.S.T. Daughtry, and D.R. Upchurch. Remote sensing for crop management. *Photogrammetric Engineering and Remote Sensing*, 69(6):647–664, 2003.

[46] P. Rigaux, M. Scholl, and A. Voisard. *Spatial Databases – With Application to GIS*. Morgan Kaufmann/Elsevier, 2001.

[47] N. Ritter and M. Ruth. The GeoTiff data interchange standard for raster geographic images. *International Journal of Remote Sensing*, 18(7):1637–1647, 1997.

[48] A. Simitsis, P. Vassiliadis, and T. Sellis. Optimizing ETL Processes in Data Warehouses. In *Proc. of the 21st Int. Conf. on Data Engineering*, pages 564–575, 2005.

[49] S. Thakkar, C. A. Knoblock, and J. L. Ambite. Quality-Driven Geospatial Data Integration. In *Proc. of the 15th Annual ACM Int. Symposium on Advances in Geographic Information Systems*, 2007.

[50] M. J. Ungerer and M. F. Goodchild. Integrating spatial data analysis and GIS: a new implementation using the Component Object Model (COM). *International Journal of Geographical Information Science*, 16(1):41–53, 2002.

[51] W. J. D. van Leeuwen, A. R. Huete, and T. W. Laing. MODIS Vegetation Index Compositing Approach: A Prototype with AVHRR Data. *Remote Sensing of Environment*, 69(3):264–280, 1999.

[52] T. C. Vance, N. Merati, S. M. Mesick, C. W. Moore, and D. J. Wright. GeoModeler: tightly linking spatially-explicit models and data with a GIS for analysis and geovisualization. In *Proc. of the 15th Annual ACM Int. Symposium on Advances in Geographic Information Systems*, 2007.

[53] Y. Xue, W. Wan, Y. Li, J. Guang, L. Bai, Y. Wang, and J. Ai. Quantitative Retrieval of Geophysical Parameters Using Satellite Data. *Computer*, 41(4):33–40, 2008.