Integrating distributed data grid, ontology and Web-based workflow technologies into geospatial cyberinfrastructure: system design and case study

Jianting Zhang

Department of Computer Science
The City College of New York
New York, NY, 10031
Email: jzhang@scs.ccny.cuny.edu

Abstract
Geospatial research increasingly relies on shared geospatial data, interconnected models and successively refined analysis which requires not only more powerful but also more accessible cyberinfrastructure systems for support. In this study, we propose to integrate data grid, ontology and Web-based workflow technologies to build more accessible cyberinfrastructure systems for geospatial computing by providing essential functionality such as Web-based workflow composition and execution, semantically enhanced data searching and workflow validation, seamless integration of data search and workflow composition and automatic data and workflow provenance. A prototype system is developed to demonstrate the feasibility of the proposed approach and key features are illustrated using an ecological case study.

1 Introduction

One of the key challenges in cyberinfrastructure research is how to make distributed data and processing units talk to each other in an efficient and effective way to help solve larger problems and answer new scientific inquires. While signficant progresses have been made on the developments and applications of geospatial clearinghouses and portals for data sharing (Masser 1999, Crompvoets et al 2004, Goodchild 2007) and high-performance geospatial computing for modeling and simulations (Wang and Liu 2009, Yang et al 2010) over the past few years, there are limited researches on the connections or lineages among geospatial data and geospatial computing which are becoming increasingly important in cyberinfrastructure and e-science (Davidson and Freire 2008, Yue et al 2010). The scientific workflow technologies provide an attractive approach to this important research topic. While adopting Web Services[1] and Semantic Web[2] technologies that are mostly designed for business data certainly is a viable approach (Sun et al 2012), we argue that alternative approaches, especially those that can handle both data intensive and computation intensive nature of geospatial computing more effectively, may suit the community needs better.

In this study, we report our work on integrating several technologies and software packages from distributed data grid and scientific workflow system domains and developing a prototype system that allows Web-based geospatial workflow editing and execution and ontology-based semantic data searching and workflow validations. More specifically, geospatial data and geospatial processing modules are annotated with Web

Ontology Language (OWL[3]) based ontologies and such annotations are attached to both data files and workflows where the iRODS data grid system[4] and Kepler scientific workflow system[5] are used, respectively. We extend Kepler by mapping between Kepler workflow processing units and ArcGIS[6] geoprocessing tools so that ArcGIS geoprocessing tools can be used in Kepler scientific workflow system for visual workflow composition and editing. Kepler is further extended to allow workflow editing in a Web environment using the Java Applet technologies and allow remote workflow executions at the server side by providing a new workflow scheduling and execution mechanism. Furthermore, by reusing Kepler's ontology based semantic validation module in a Web environment, we are able to check both structural and semantic compatibilities among data and workflow processing units as well as the compatibilities among connecting workflow processing units before executions in a production mode. The lineages among the input data, processing units and output data are thus well documented using workflow technologies.

When compared with the state-of-the-art in geospatial portals and geospatial workflows, we argue that our architectural design has quite a few advantages. First, our design seamlessly integrates data and its processing pipeline in a workflow environment. Unlike geospatial portals that are mostly designed for data searching and visualization only, the ontology-based search results on geospatial data stored in standalone or federated iRODS data grid systems can be visualized as workflow components (data sources in this case) and used immediately in workflow compositions through drag-and-drop based interactive editing. Second, unlike Web-service based semantic workflow composition and validation, our design does not require publishing every geospatial dataset as a W3C WSDL[7] web service before they can be used. It is not always efficient or even possible to publish geospatial data as WSDL web services as serialization and deserialization are both complex and expensive, especially when the data volumes are large. On the other hand, attaching the ontology-based annotations as part of metadata of a geospatial dataset can be simple and flexible. Third, our design makes coarse-grained geospatial modeling easy by supporting interactive drag-and-drop based workflow composition and editing which is much easier than calling APIs of data access and processing modules using programming or scripting languages from a user perspective. This essentially eliminates or significantly decreases the requirements on programming skills imposed on scientist so that they can focus on their domain problems – an ultimate goal of cyberinfrastructure research and development. This feature is further enhanced by ontology based semantic validations in our prototype system. Fourth, our design allows users to access datasets, compose/edit processing pipelines and visualize outputs, all through a Web-based interface. We note that while geospatial portals usually provide Web-based interfaces for searching and data visualization, many cyberinfrastructure systems still only provide text-based console interfaces. Cyberinfrastructure systems with Web-enabled frontends are more desirable to end users.

As a case study, we will demonstrate our system design and implementation using an ecological example. The following key features in the prototype system are highlighted: (1) Web-based geospatial workflow composition, editing and execution (2) searching on an iRODS data grid system and using search results as workflow data sources through drag-and-drop (3) ontology based workflow validations over the Web (4) processing units in validated geospatial workflows are automatically mapped to ArcGIS Geoprocessing tools for remote execution.

The rest of the chapter is arranged as follows. Section 2 introduces background and motivations of the proposed research and development efforts and overviews the key technologies in realizing the prototype system. Section 3 presents the system architecture and the implementation details of key components. Section 4 is the case study to demonstrate the features of the prototype system. Section 5 provides discussions on integrating the prototype system with both cluster computers and Graphics Processing Units (GPUs) based high-performance computing resources. Finally Section 6 is the summary and conclusions.

2 Background, Motivations and Related Technologies

Like many scientific disciplines, geospatial research is increasingly becoming data intensive and computing driven with respect to data collection, exploratory study, model simulation and decision support. Unlike traditional research that individual researchers are in charge of the whole life cycle of data acquisition, analysis and publication, modern geospatial research relies on shared geospatial data, interconnected models and successively refined analysis. Data, models and analysis are distributed across research groups both logically and physically. However, the linkage among data and models become crucial to understand the complex analysis pipelines and perform new scientific inquiries. In recent years, a few cyberinfrastructure techniques, such as data grid, metadata/semantics, high-performance computing and scientific workflow, have been developed to make large-scale scientific research easier.

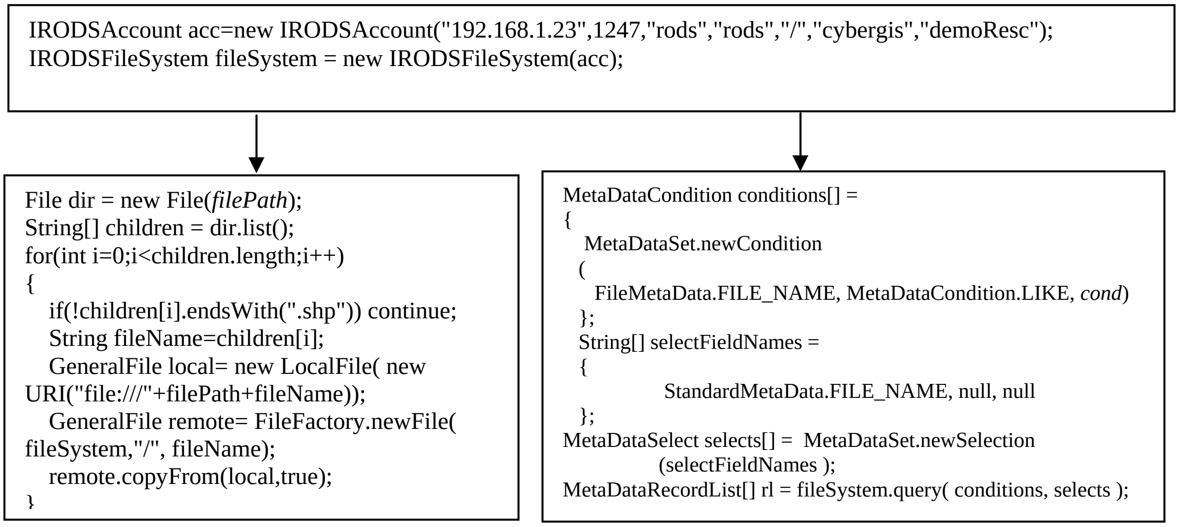
Data management has been an important discipline in geospatial computing and many technologies, such as geospatial data clearinghouse, Spatial Data Infrastructure (SDI) and geospatial portal, have been developed to allow efficient data searching and sharing in the past few decades. When the searching results are linked with a GIS, the related data can be easily visualized. In addition, in recent years, the Semantic Web technologies have been integrated with these traditional geospatial data management technologies to allow semantics based data searching, data discovery and data provenance (Yue et al 2011). While these technologies are certainly useful in their targeted application domains, a major drawback is that geospatial data are separated from their processing pipelines and the linkage among data and models are not kept. In addition to lacking automatic provenance, from a system perspective, several features that are desirable for large-scale geospatial data processing, such as online accesses to archived data, interactive drag-and-drop based visual modeling and on-demand accelerations by high-performance computing resources, are not supported by traditional geospatial data management techniques.

These shortcomings have motivated us to develop a set of technologies that provide end-to-end support for modern geospatial computing in the context of CyberGIS research (Wang et al 2012) by integrating leading cyberinfrastructure technologies. More specifically, we propose to use semantics-enhanced data grid technologies to store, query and online access interconnected geospatial datasets, and, scientific workflow technologies for visual modeling, automatic provenance and distributed execution. While the proposed approach can be applied in different computing environments, currently we are targeting at the Web-based computing for easy accesses to the Graphics User

Interfaces (GUIs) of the prototype system, such as searching on distributed data grids, workflow composition and execution and ontology-based semantic validation of workflows. Before we present the system design and implementations in Section 3, we next briefly introduce the relevant key cyberinfrastructure technologies, including data grid and scientific workflow related ones.

2.1 Data Grid and iRODS

According to (Allcock et al 2005), a Data Grid is an architecture or set of services that enable individuals or groups of users the ability to access, modify and transfer extremely large amounts of geographically distributed data for research purposes. Compared with file systems on single computers, data files in a data grid system are replicated and distributed according to certain policies for both fast accesses and fault tolerances. Usually a metadata catalog is maintained for one or more data grid systems to store metadata associated with the data files in the data grid systems and keep track of data movements and replications. By querying the metadata catalog, data files can be discovered and subsequently accessed either interactively or programmatically through APIs. Data grid technologies are closely related to distributed parallel file systems such as IBM General Parallel File System (GPFS[8]) in the commercial sector which is fairly expensive for small research groups. In recently years, an open source data grid system iRODS (*i* Rule Oriented Data Systems), that originates from the Storage Resource Broker (SRB[9]) system, has been widely adopted in the scientific research communities, ranging from plant genomics to nuclear research[10]. iRODS has also been adopted by NASA for data dissemination (Schnase et al 2012) and extensively used to manage larges-scale environmental model results[11]. However, in these geospatial applications, iRODS was mostly used as a data achieving tool and has not been used to support interactive geospatial modeling.



```
IRODSAccount acc=new IRODSAccount("192.168.1.23",1247,"rods","rods","/","cybergis","demoResc");
IRODSFileSystem fileSystem = new IRODSFileSystem(acc);
```

```
File dir = new File(filePath);
String[] children = dir.list();
for(int i=0;i<children.length;i++)
{
   if(!children[i].endsWith(".shp")) continue;
   String fileName=children[i];
   GeneralFile local= new LocalFile( new
URI("file:///"+filePath+fileName));
   GeneralFile remote= FileFactory.newFile(
fileSystem,"/", fileName);
   remote.copyFrom(local,true);
}
```

```
MetaDataCondition conditions[] =
{
   MetaDataSet.newCondition
  (
    FileMetaData.FILE_NAME, MetaDataCondition.LIKE, cond)
  };
  String[] selectFieldNames =
  {
             StandardMetaData.FILE_NAME, null, null
  };
MetaDataSelect selects[] =  MetaDataSet.newSelection
             (selectFieldNames );
MetaDataRecordList[] rl = fileSystem.query( conditions, selects );
```

Fig. 1 Illustration of Programmatically Accessing iRODS for Data Uploading and Metadata Searching Using Jargon Java APIs

Since our interests in using data grid technologies in general and iRODS in specific are to support semantically enriched queries and interactive visual modeling, we need to programmatically access iRODS data grids to populate/search metadata and retrieve/upload/replicate data files. We have used the Jargon java API[12] provided by iRODS for this purpose. Fig. 1 lists a few code segments to demonstrate how to create an iRODS account, upload data and query metadata to help understand how the system works. In a way similar to querying metadata, we can also add or modify metadata entries in the form of (name, value, unit) triplets. The mechanism allows attaching workflows as metadata entries for provenance purposes on arbitrary datasets.

2.2 Scientific Workflow System and Kepler

The importance of Scientific Workflow (SWF) has been well recognized in the development of cyberinfrastructure (Yu and Buyya 2005, Deelman et al 2009). Quite a few SWF systems have been developed to facilitate composition, scheduling and execution of diverse processing pipelines (Yu and Buyya 2005, Deelman et al 2009). Among many features of SWF systems, the following ones are especially desirable for distributed geospatial computing. First, SWF systems usually provide mechanisms to wrap around heterogeneous processing units written in different programming languages and hosted by different systems and platforms, and, provide unified interfaces for invocations by the SWF systems. Second, some SWF systems provide graphic user interfaces for visual programming through drag-drop-connect on iconized processing units. The functionality is very similar to ESRI ModelBuilder[13] embedded in ArcGIS that has been popular in the geospatial computing community. While many existing SWF systems provide such visual programming interface in a desktop computing environment, there are a few attempts to port the functionality in a Web environment (Tuot et al 2008, Sun et al 2012) with limited successes due to technological complexities. Third, validated workflows naturally represent the lineages among different datasets and can be used for provenance purposes.

Kepler (Ludäscher et al 2006) is one of the leading SWF systems and has also been widely in many application domains. Kepler is built on top of the Ptolemy II system dated back to early 2000s based on the previous developments at UC Berkeley[14]. Ptolemy II is written in Java and uses a Java software infrastructure called Diva[15] to render workflow components (or directors, actors, ports and parameters in Ptolemy II terminologies) and interact with users. While Ptolemy II is primarily developed for modeling, simulation, and design of concurrent, real-time, embedded systems, many of its features meet the requirements of scientific workflow systems. As such, Ptolemy II was chosen as the base for the development of Kepler scientific workflow system (Ludäscher et al 2006). However, as Kepler is designed to be a desktop system and is too voluminous for Web applications (by using Java Web Start technology[16]), in this study, we use Ptolemy II directly for workflow composition and scheduling. We re-use the Web-based geospatial workflow system (herein referred as WGWFS) that we have developed previously (Zhang 2012) with the following extensions. First, the ontology-based semantic validation module in desktop Kepler has been imported to WGWFS. Second, the semantics-enhanced WGWFS is integrated with the iRODS data grid system to allow storing, searching and selecting data files in a drag-and-drop manner. Third, the composed geospatial workflows can be stored in iRODS as part of the metadata for relevant geospatial datasets for provenance purposes. The integration of the iRODS data grid and the Kepler workflow system provides several desired features for CyberGIS applications. While the architectural design and the implementation details of the new system will be presented in Section 3, we refer to (Zhang 2012) for more technical details on the development of WGWFS, including introductions to the most relevant features in Ptolemy II and Kepler and discussions on the related work on Web-based geospatial workflow systems.

3 System Architecture and Implementations

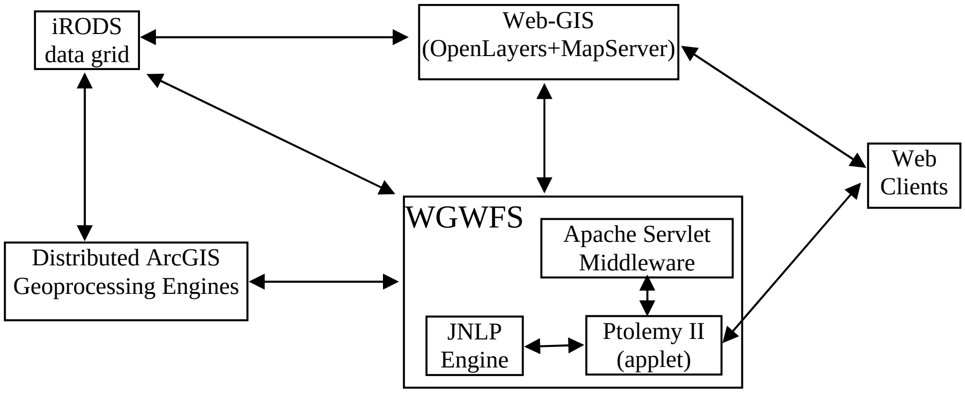

iRODS data grid
Web-GIS (OpenLayers+MapServer)
Web Clients
WGWFS
Apache Servlet Middleware
Distributed ArcGIS Geoprocessing Engines
JNLP Engine
Ptolemy II (applet)

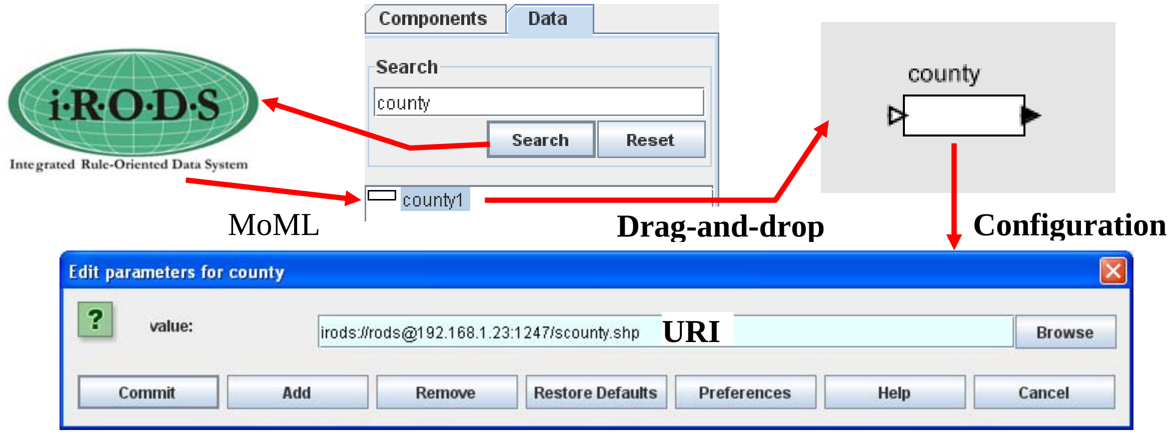
Fig. 2 High-Level System Architecture

The overall system architecture is shown in Fig. 2. The Web clients first invoke the WGWFS workflow composition environment (Zhang 2012) implemented on top of Ptolemy II as a Java applet using the Java Network Launching Protocol (JNLP[17]). The applet communicates with both the iRODS data grid system and the distributed ArcGIS Geoprocessing engines through the middleware components implemented as Java Servlets hosted in an Apache Web server[18]. The Web clients can search the data grids (c.f. Fig. 1) through a graphics user interface in the workflow applet which delegates the searching to the corresponding Servlet middleware on the server side before the data queries are actually performed by the iRODS data grid system. The Servlet middleware is also responsible for transforming the binary query results into a representation using the internal workflow language called MoML (Modeling Markup Language) in Ptolemey/Kepler (Lee and Neuendorffer 2000, Ludäscher et al 2006). Since Ptolemey/Kepler is able to visualize the workflow components (data sources in this case) expressed in MoML, the query results can be used as data sources in Ptolemey/Kepler in a drag-and-drop manner. Similarly, the composed workflow can also be stored in the iRODS data grid system either as data files or as metadata to be associated with certain data files.

A separate Servlet middleware is responsible for generating a workflow execution plan based on the composed workflow, mapping the workflow processing units to appropriate ArcGIS geoprocessing tools and executing the geoprocessing tools on a single machine or multiple distributed machines equipped with ArcGIS geoprocessing engines (Zhang 2012). A summary reporting the workflow execution status with links to the inputs, intermediate results and final outputs is returned to the workflow applet. Since the summary report is composed in standard HTML, the geospatial data corresponding to the relevant datasets can be visualized by a Web-GIS (currently implemented using the open source GDAL[19]+MapServer[20]+OpenLayers[21] software stack) by following the links (Zhang 2012). If the intermediate and final outputs of the ArcGIS geoprocessing tools are stored in the iRODS data grid system, then the WebGIS will connect the data grid system to retrieve the relevant geospatial data (e.g., in the ESRI shapefile format) and publish them as Open Geospatial Consortium (OGC[22]) Web services so that they can be conveniently and efficiently visualized in a variety of Web-based mapping systems (e.g., OpenLayers and ArcGIS Web APIs[23]). We next provide the implementation details of the two newly added modules, i.e., integration with data grid and semantic validation of workflows. A case study will be provided in Section 4 for illustration purposes.

3.1 Integration with data grid

Similar to the workflow execution middleware discussed in (Zhang 2012), the functionality in integrating the Web-based geospatial workflow system with the iRODS data grid system is also implemented as a Java Servlet hosted in the Apache Web server that also hosts the workflow applet. This is primarily because the "same-origin" restriction imposed to Java applets with respect to data communication with remote systems. As mentioned earlier, the Servlet middleware has two roles. The first role is to delegate the metadata searching requests from the workflow applet to the remote iRODS data grid system. The second role is to convert the binary query results from the iRODS system to the MoML format so that each query result can be visualized (iconized) and used in workflow composition through drag-and-drop.



value: irods://rods@192.168.1.23:1247/scounty.shp

Fig. 3 Searching Data in Remote Data Grid System for Visual Workflow Composition

In a way similar to integrating semantic web technologies and geospatial catalog services for geospatial information discovery (Yue et al 2011), our design enhances traditional keyword matching based searching with semantics by exploring the inheritance and association relationships that are typically caught in OWL based ontologies which can be authorized in quite a few tools such as Protégé[24] . While more details on the geospatial semantics will be discussed in Section 3.2, our current implementation on the semantically enhanced search mostly focuses on concept expansion. For example, if a keyword search matches a particular OWL concept, then the ancestor and descendant concepts as well as other associated concepts will also be matched. Since querying OWL documents using the standard RDF query language SPARQL[25] is quite expensive, our design extracts the inheritance and association relationships from the OWL documents and store them in main memory data structures as hash tables so that the concept expansion process can be realized more efficiently and achieve real-time responses even for large ontologies.

3.2 Semantic validation of workflows

Semantic validation is an integral part of the desktop-based Kepler SWF system and has been used in validating geospatial workflows based on both structural and semantic compatibilities in geospatial applications (Zhang et al 2005, Zhang 2006). Structural data types describe representational aspects of data (Bowers and Ludäscher 2005). The type lattice of Ptolemy II defines a partially ordered set of data types (Zhao et al 2010). Types can be statically declared or left to be resolved during execution by imposing type constraints (Xiong and Lee 2000). The Ptolemy type lattice provides both base data types (such as Boolean, integer, double) and collection data types (such as array, matrix, record). Kepler allows annotate the input and output parameters of workflow processing units based on one or more ontologies, i.e., assigning semantic data types by describing the conceptual aspects of data (Bowers and Ludäscher 2005).

We use semantic data types to symbolically check the compatibilities among data of different semantic types. For example, a particular workflow data communication channel (the connection between two processing units) may be structurally compatible (e.g., one processing unit produces a record type consumed by another processing unit), but semantically incompatible (because the data conceptually denotes a different type of entity). The left part of Fig. 4 shows one of the ontologies that we have used to assign semantic types of workflow processing units which corresponds to ArcGIS ArcObject[26] class hierarchies closely. Among the classes that are most relevant to the case study in Section 4, *Featureclass* extends *Table* by adding geometric components of data and can be *PointFeature*, *PolylineFeature* or *PolygonFeature*. Note that the properties of classes are not shown in the figure. We use the semantic data types derived from the data type ontology to annotate the inputs and outputs of workflow processing units. While more complex rules to determine the “soft” compatibilities based on both class and property labels are being developed, we currently adopt a simple rule to determine the “hard” compatibility between the output of a preceding workflow: if the ontological concept assigned to the output of the preceding workflow processing uint is the same as or a subclass of the ontological concept assigned to the input of the connecting workflow processing unit then they are semantically compatible; otherwise not. Using the same semantic web technology, we can also annotate the classifications of the workflow processing units themselves by following the organization hierarchy of ArcGIS geoprocessing tools as shown in the right part of Fig. 4.

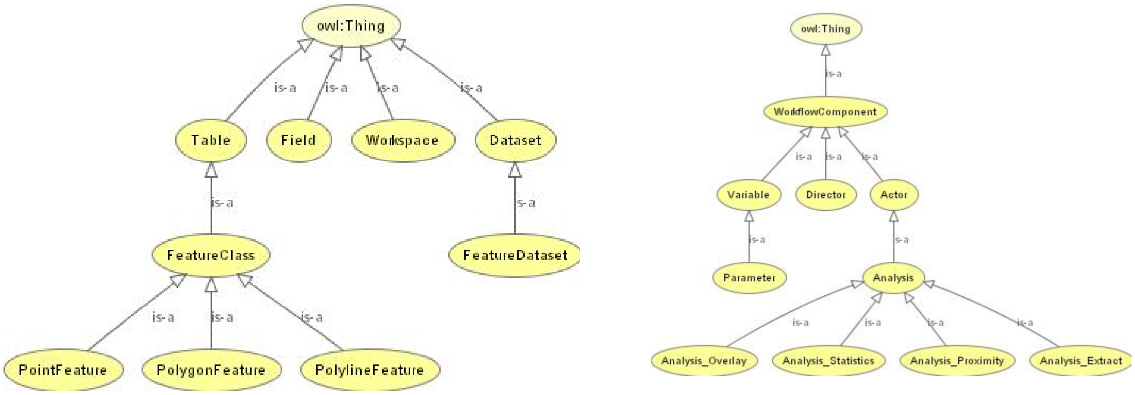


Fig. 4 Ontologies (Partial) of Semantic Data Types and Workflow Components

In this study, we have reengineered the semantic validation component in desktop Kepler and made it compatible with WGWFS so that semantic validation can be performed in a Web computing environment. As will be demonstrated in Section 4, the Web-based semantic validation system includes two major graphics user interfaces: one is invoked from the context menu of workflow process units to annotate the structural and semantic data types of the input and output parameters of the respective workflow processing units, and, one is to check both the structural and semantic compatibilities among the workflow data communication channels. The validated geospatial workflows can be associated with the data sources, intermediate results or data sinks for provenance purposes.

4 An Ecological Case Study

We use a tutorial example provided by the Department of Biological Sciences at the University of Alberta for its GIS in Ecology class[27] on calculating the area of burned riparian forest from a few input datasets after applying some typical geospatial operations, such as buffer, union, clip and intersect. More specifically, the example buffers water bodies and streams, unions the buffers to derive the riparian buffer, clips forestry with the riparian buffer to derive riparian forestry and intersects riparian forestry with the fire history before finally calculating the area of burned riparian forest. The geospatial computing task was originally implemented in ArcGIS ModelBuilder in a desktop computing environment. We will demonstrate how our prototype system implements the same geospatial computing task in a CyberGIS computing environment and how integrating data grid, ontology and Web-based geospatial workflow technologies can help evolving traditional desktop GIS into CyberGIS. Fig. 5 shows the composed geospatial workflow to provide an overview of the geospatial datasets and operations that are involved in the processing pipeline. We note that the workflow is conceptually similar to the model developed in ArcGIS ModelBuilder in the tutorial but the workflow can be composed in a Web environment and can be executed in distributed computing environments without requiring ArcGIS programming skills.

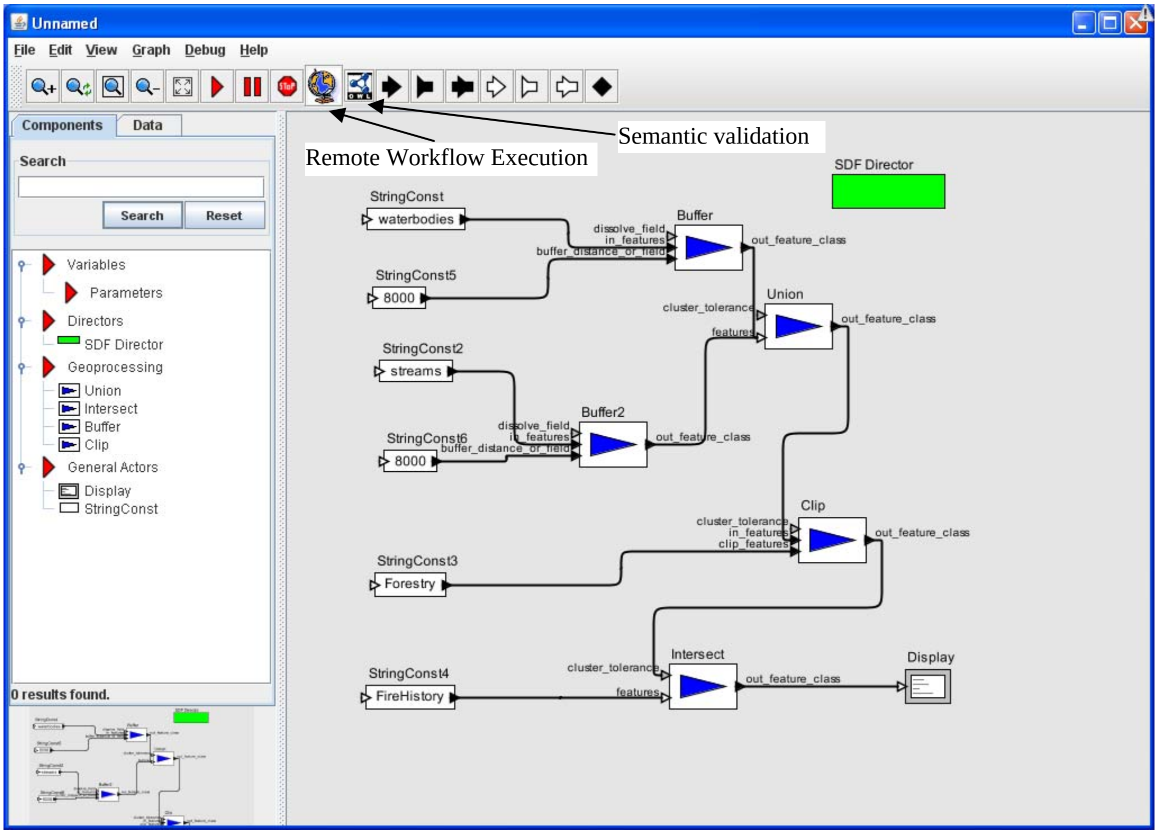


Fig. 5 Composed Geospatial Workflow for the Ecological Case Study

After an iRODS data grid system has been setup, we can use the iRODS Jargon APIs to upload all relevant datasets, together with their metadata, into the data grid system (c.f. Fig. 1). Semantic data types are assigned to the datasets and stored as metadata. For example, the *streams* dataset are annotated as *PolylineFeature* while *watherbodies*, *Forestry* and *Firehistory* datasets are annotated as *PolygonFeature*, according to the ontology shown in the left part of Fig. 4. When users query datasets that have a semantic data type of *PolylineFeature*, only *streams* will be returned. However, when users query datasets that have a semantic data type of *FeatureClass*, all of the four datasets will be returned due to the ontological hierarchy.

As shown in Fig. 3, users can search the desired geospatial datasets from within the workflow composition environment and the search results are represented as iconized workflow components for subsequent drag-and-drop based workflow composition. This is quite different from many geospatial portals that only search data catalogs. The tight integration between data and its metadata in the iRODS data grid system largely avoids the problem of broken links between data and metadata. In addition, as a distributed file system, data stored in the iRODS data grid system can be programmatically accessed conveniently without requiring publishing the data as Web services which is not always efficient or even possible. More importantly, the search results from the data grid system can be directly used in subsequent modeling (other than standard visualization), a feature that is lacking by many geospatial portal systems.

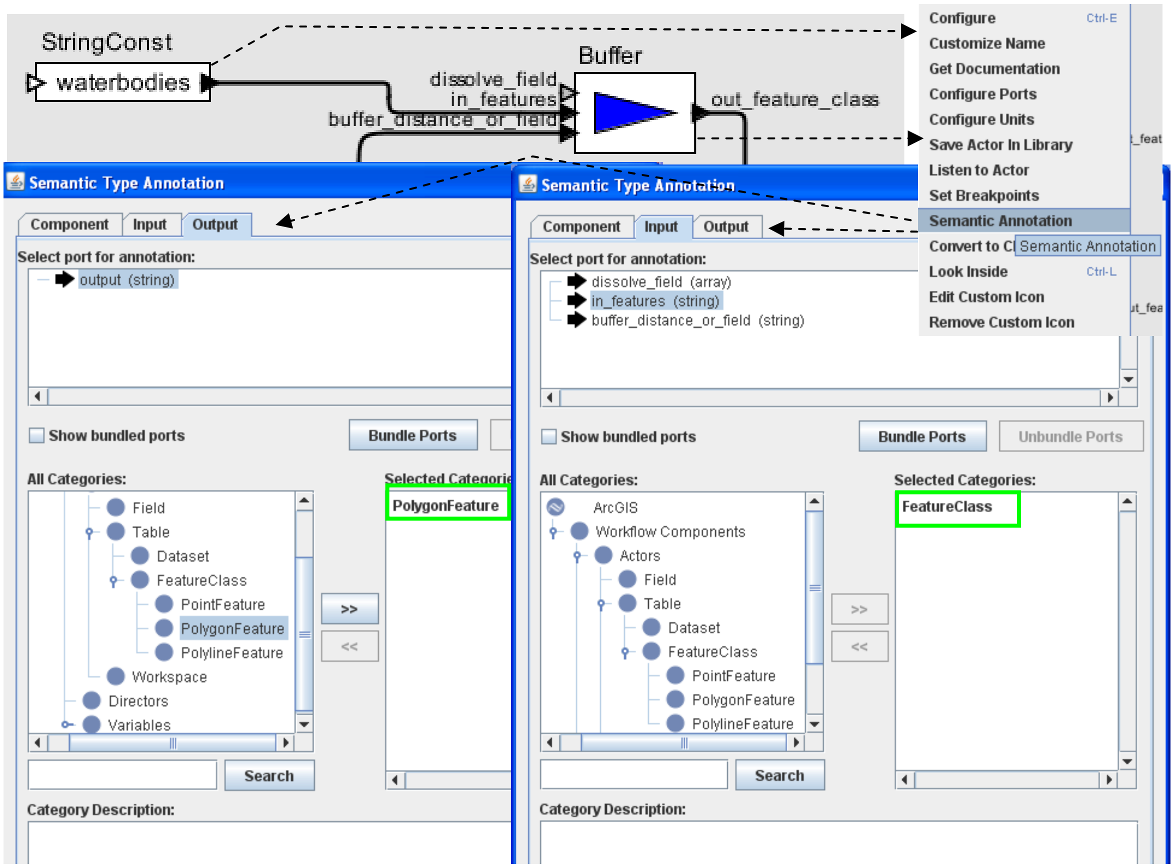
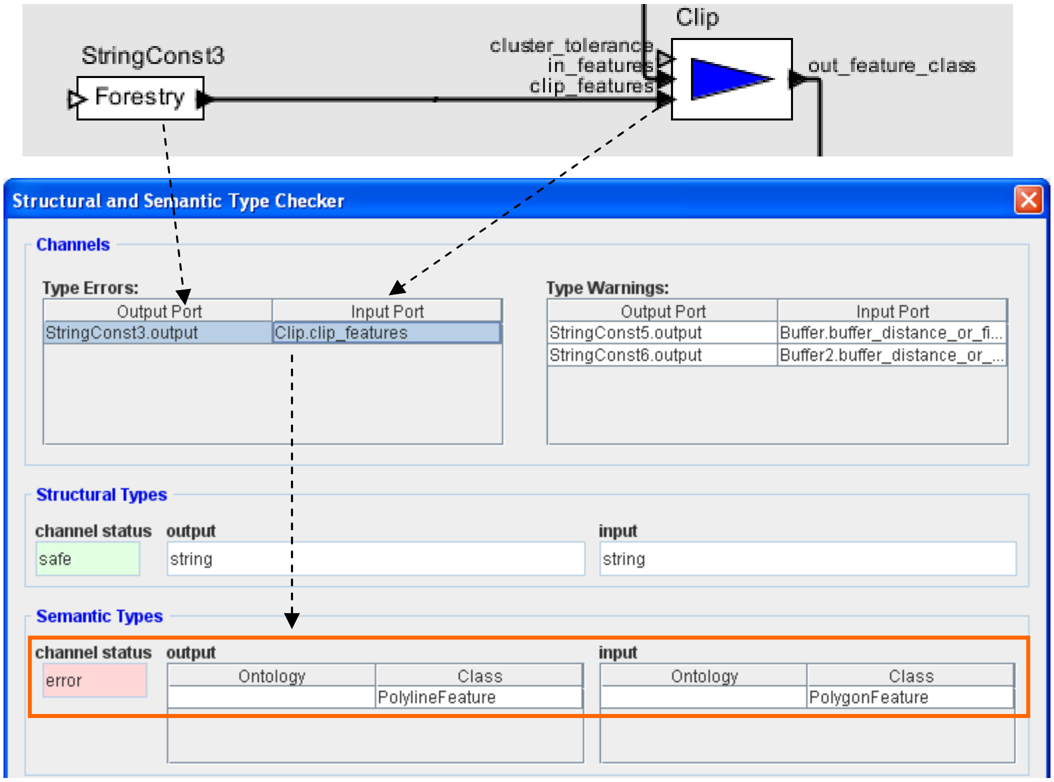


Fig. 6 Illustration of Compatible Semantic Data Types: The output of the data source (*watherbodies*) has a semantic type of *PolygonFeature* and the input of the *Buffer* processing tool has a semantic type of *FeatureClass* and they are compatible

The ontology based semantic validation is illustrated in Fig. 6 and Fig. 7 for compatible and incompatible semantic data types, respectively. In Fig. 6, for the data communication channel between the data source (*waterbodies*) and the *in_feature* input of the *Buffer* processing tool, the data source (with only one output) has a *PolygonFeature* semantic type and the *in_feature* has a *FeatureClass* semantic type. Since *PolygonFeature* is a subclass of *FeatureClass*, the connection is considered to be semantically compatible and requires no further action by the system. On the other hand, as shown in Fig. 7, for the data communication channel between the data source (*forestry*) and the *clip_feature* of the *Clip* processing tool, the data source has a *PolylineFeature* semantic type while the input of the *Clip* tool expects a *PolygonFeature* semantic type and they are not compatible. The error is captured by the workflow system and reported in a summary table shown in the top-left part of Fig. 7.



Structural and Semantic Type Checker

Channels

Type Errors:

Output Port	Input Port
StringConst3.output	Clip.clip_features

Type Warnings:

Output Port	Input Port
StringConst5.output	Buffer.buffer_distance_or_fi...
StringConst6.output	Buffer2.buffer_distance_or_...

Structural Types

channel status	output	input
safe	string	string

Semantic Types

channel status: error

output

Ontology	Class
	PolylineFeature

input

Ontology	Class
	PolygonFeature

Fig. 7 Illustration of Incompatible Semantic Data Types: the input of the *Clip* processing unit requires a *PolygonFeature* semantic type while the output of the data source (*Forestry*) has a *PolyglineFeature* semantic type.

With respect to workflow execution, we refer to (Zhang 2012) for technical details on mapping the workflow processing units to geoprocessing tools and execute composed geospatial workflows on remote servers equipped with ArcGIS Geoprocessing Engines from within the workflow composition environment (c.f. Fig. 5). We note that it is relatively straightforward to integrate ArcGIS Geoprocessing tools with iRODS data grid system by utilizing their Java APIs. A more elegant solution is to develop a customized ArcGIS Workspace class directly on top of the iRODS data grid system so that data files stored in iRODS can be more efficiently streamed among the two systems. This is left for future work.

5 Discussions on the Extensions to HPC Environments

Our existing work has primarily focused on the traditional multi-tier client-server computing environments for Web-enabled geospatial workflow composition and execution. While the current prototype system is capable of utilizing multiple distributed ArcGIS geoprocessing engines on either Windows or Linux machines, it has not been designed to utilize high-performance computing resources in the cyberinfrastructure, especially in the context of parallel geoprocessing that has gained increasing popularities in CyberGIS applications. Our proposed extensions have the following two aspects. First, a Java Servlet middleware can be developed to automatically transfer data from local file systems and iRODS data grid system to the targeted cyberinfrastructure and automatically submit jobs based on the composed geospatial workflows. Algorithms in parallelizing the workflow processing units can be further developed to improve overall performance. Second, for geospatial computing tasks that are both data and computing intensive and/or require real time interactions, if the major processing units in the workflows are data-parallelizable, then the composed workflows can be shipped to machines that host the data and equipped with Graphics Processing Units (GPUs). General Purpose computing on GPUs (GPGPU[28]) technologies can be used to accelerate geospatial computing (Zhang 2010). Our CudaGIS prototype system (Zhang and You 2012) can potentially replace ArcGIS Geoprocessing tools for certain geospatial computing tasks and gain signficant speedups (10-40X for in-memory systems and 3-4 orders for disk-resident systems). Note that when cluster computers are equipped with GPUs, which is increasingly become available from both institutional grid computing resources and commercial cloud computing resources, the two aspects can actually be merged to further allow solving larger scale geospatial problems. We believe the workflow, ontology and data grid technologies discussed in this chapter are orthogonal to both cluster-based and GPU-based high-performance computing and their integrations will make geospatial cyberinfrastructure not only more powerful but also more usable to facilitate scientific inquires.

Summary and Conclusions

Data grids, ontologies and scientific workflow technologies are the building blocks of cyberinfrastructure. While these technologies have been applied to geospatial data management and computing in different ways, the integration of these technologies to allow seamless and semantically enhanced data search, workflow composition, validation and execution has not been extensively explored previously. We consider the proposed approach a step forward towards more accessible geospatial cyberinfrastructure development. The functionality is illustrated through an ecological example in the case study. In addition to be able to tightly associate the provenance information of geospatial datasets by treating workflows as metadata, we have also discussed application scenarios in using cluster computer based and GPU based high-performance computing resources for parallel geospatial computing so that the workflows can also be used to facilitate parallelization.

In conclusion, despite signficant development efforts may be required, tightly integrating data grids, ontologies and scientific workflow technologies in a cyberinfrastructure environment is not only feasible but also promising for geospatial computing. In addition to applying the prototype system to solve more domain-specific geospatial problems (including ontology development), our future work focus will be on integrating the prototype system with high-performance computing resources in different types of cyberinfrastructure systems.

ACKNOWLEDGEMENT

This work is supported in part through a PSC-CUNY grant (64698-00 42).

References

1. Allcock, B., Chervenak, A., Foster, I., Kesselman, C. and Livny, M. (2005). Data Grid tools: enabling science on big distributed data. Journal of Physics: Conference Series 16 571.
2. Bowers, S. and Ludäscher, B. (2005). Actor-Oriented Design of Scientific Workflows. International Conference on Conceptual Modeling (ER), Springer LNCS, 3716: 369-384
3. Crompvoets, J., Bregt, A., Rajabifard, A. and Williamson, I. (2004). Assessing the worldwide developments of national spatial data clearinghouses. International Journal of Geographical Information Science 18(7): 665-689.
4. Davidson, S. B. and Freire, J. (2008). Provenance and scientific workflows: challenges and opportunities. Proceedings of the ACM SIGMOD Conference.
5. Deelman, E., Gannon, D., Shields, M. and Taylor, I. (2009). Workflows and e-Science: An overview of workflow system features and capabilities. Future Generation Computer Systems 25(5): 528-540.
6. Goodchild, M. F., Fu, P. and Rich, P. (2007). Sharing Geographic Information: An Assessment of the Geospatial One-Stop. Annals of the Association of American Geographers 97(2): 250-266.
7. Lee, E. A. and Neuendorffer, S. (2000). MoML - A Modeling Markup Language in XML (Version 0.4). Technical Memorandum UCB/ERL M00/12. http://ptolemy.eecs.berkeley.edu/publications/papers/00/moml/
8. Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E. A., Tao, J. and Zhao, Y. (2006). Scientific workflow management and the Kepler system. Concurrency and Computation: Practice and Experience 18(10): 1039-1065.
9. Masser, I. (1999). All shapes and sizes: the first generation of national spatial data infrastructures. International Journal of Geographical Information Science 13(1): 67-84.
10. Schnase, J. L., Tamkin, G. S., Ripley, W. D. I., Stong, S., Gill, R. and Duffy, D. Q. (2012). The Virtual Climate Data Server (vCDS): An iRODS-Based Data Management Software Appliance Supporting Climate Data Services and Virtualization-as-a-Service in the NASA Center for Climate Simulation. NASA Report. http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20120009334_2012009166.pdf
11. Sun, Z., Yue, P. and Di, L. (2012). GeoPWTManager: a task-oriented web geoprocessing system. Computers and Geosciences 47: 34-45.
12. Tuot, C. J., Sintek, M., et al. (2008). IVIP --- A Scientific Workflow System to Support Experts in Spatial Planning of Crop Production. Proceedings of the International Conference on Scientific and Statistical Database Management (SSDBM).
13. Wang, S., Wilkins-Diehr, N. R., and Nyerges, T. L. 2012. CyberGIS – Toward Synergistic Advancement of Cyberinfrastructure and GIScience: A Workshop Summary. Journal of Spatial Information Science, 4: 125-148.
14. Wang, S. W. and Liu, Y. (2009). TeraGrid GIScience Gateway: Bridging cyberinfrastructure and GIScience. International Journal of Geographical Information Science 23(5): 631-656.

15. Xiong, Y. and Lee, E. A. (2000). An Extensible Type System for Component-Based Design. International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Springer LNCS 1785: 20-37
16. Yang, C. W., Raskin, R., Goodchild, M. and Gahegan, M. (2010). Geospatial Cyberinfrastructure: Past, present and future. Computers Environment and Urban Systems 34(4): 264-277.
17. Yu, J. and Buyya, R. (2005). A taxonomy of scientific workflow systems for grid computing. ACM SIGMOD Record 34(3): 44-49.
18. Yue, P., Gong, J., Di, L., He, L. and Wei, Y. (2011). Integrating semantic web technologies and geospatial catalog services for geospatial information discovery and processing in cyberinfrastructure. GeoInformatica 15(2): 273-303.
19. Yue, P., Gong, J. and Di, L. (2010). Augmenting geospatial data provenance through metadata tracking in geospatial service chaining. Computers and Geosciences 36(3): 270-281.
20. Zhang, J. (2012). A Practical Approach to Developing a Web-based Geospatial Workflow Composition and Execution System. Proceedings of Com.Geo Conference.
21. Zhang, J. and You, S. (2012). CudaGIS: Report on the Design and Realization of a Massive Data Parallel GIS on GPUs. Technical Report. http://www-cs.ccny.cuny.edu/~jzhang/papers/cudagis_tr.pdf
22. Zhang, J. (2010). Towards Personal High-Performance Geospatial Computing (HPC-G): Perspectives and a Case Study. Proceedings of ACM HPDGIS Workshop.
23. Zhang, J. 2006. Ontology-Driven Composition and Validation of Scientific Grid Workflows in Kepler: a Case Study of Hyperspectral Image Processing. Proceedings of International Conference on Grid and Cooperative Computing (GCC) workshops.
24. Zhang, J., Pennington, D. Michener, W. (2005). Validating compositions of geospatial processing Web services in a scientific workflow environment. Proceedings of IEEE International Conference on Web Services (ICWS).
25. Zhao, Y., Xiong, Y., et al. 2010. The design and application of structured types in Ptolemy II. International Journal of Intelligent Systems 25(2), 118-136.

[1] http://en.wikipedia.org/wiki/Web_service
[2] http://en.wikipedia.org/wiki/Semantic_Web
[3] http://en.wikipedia.org/wiki/Web_Ontology_Language
[4] https://www.irods.org/
[5] https://kepler-project.org/
[6] http://www.esri.com/software/arcgis
[7] http://www.w3.org/TR/wsdl
[8] http://www-03.ibm.com/systems/software/gpfs/
[9] http://www.sdsc.edu/srb
[10] https://www.irods.org/index.php/IRODS_User_Group_Meetings
[11] https://www.irods.org/index.php/iRODS_User_Group_Meeting_2012
[12] https://www.irods.org/index.php/Jargon
[13] http://www.esri.com/industries/defense/modelbuilder.html
[14] http://ptolemy.eecs.berkeley.edu/ptolemyII/
[15] http://embedded.eecs.berkeley.edu/diva/
[16] http://en.wikipedia.org/wiki/Java_Web_Start
[17] http://docs.oracle.com/javase/tutorial/deployment/deploymentInDepth/jnlp.html
[18] http://httpd.apache.org/
[19] http://www.gdal.org/
[20] http://mapserver.org/

[21] http://openlayers.org/
[22] http://www.opengeospatial.org/
[23] http://resources.arcgis.com/content/web/web-apis
[24] http://protege.stanford.edu/
[25] http://www.w3.org/TR/rdf-sparql-query/
[26] http://help.arcgis.com/en/sdk/10.0/arcobjects_net/ao_home.html
[27] http://www.biology.ualberta.ca/facilities/gis/uploads/instructions/2_AVD.pdf
[28] http://en.wikipedia.org/wiki/GPGPU