Developing an Ontology and ACL in an Agent-based GIS*

James J. Nolan
Center for Image Analysis
Dept. of Computer Science
George Mason University
Fairfax, VA 22030
jnolan@gmu.edu

Robert Simon
Center for Image Analysis
Dept. of Computer Science
George Mason University
Fairfax, VA 22030
simon@cs.gmu.edu

Arun K. Sood
Center for Image Analysis
Dept. of Computer Science
George Mason University
Fairfax, VA 22030
asood@gmu.edu

ABSTRACT

We present an ontology and agent communication language (ACL) for an agent-based Geographic Information System (GIS). The ontology is comprised of three critical components necessary to exchange information between GIS-domain agents, systems, or organizations, including: vector, raster, and image data; algorithms descriptions including name, inputs, outputs, and required parameters; and query/result information. Our ACL uses the ontology as the foundation of communication between agents. The ontology and ACL have been encoded in the Resource Description Framework (RDF) using the eXtensible Markup Language (XML). The ontology and ACL are central to agent communication within a distributed, multi-agent, GIS that we have prototyped using various COTS GIS software packages.

Keywords

ontology, geographic information systems, service-based architectures

1. INTRODUCTION

Geographic Information Systems (GIS) are used in the fields of cartography, resource management, real-estate, and intelligence gathering, among others. A GIS provides the means to process vector, raster, and image data and distill it into meaningful information for a particular domain. At its core, a GIS is used in the decision-making process, processing data to answer questions such as: "What is the optimal route between Washington and New York?", or "What are the expected wheat harvests this summer in Kansas?", or "Is the XYZ chemical plant producing weapons of mass destruction?".

To answer such questions requires the analysis of many different data types: vector features representing roads and land classification, raster grids representing 3-dimensional elevation data sets, and image data captured at various resolutions from space and airborne remote sensing platforms. This data is captured in varying data representations, formats, scales, and spatial resolutions.

Analysis of the data is done through the use of various algorithms. As diverse data types and algorithms have emerged, algorithm, system, and data interoperability has become problematic [3]. Many efforts have been undertaken to address this problem [1, 7] and two general approaches have emerged: (1) *Data standardization*, where GIS users agree on standard data formats, increasing data interoperability, and (2) *Algorithm interface standardization* where GIS developers agree on standard algorithm interfaces, increasing system interoperability.

These two approaches have shortfalls and have largely failed to date. There is simply too much useful legacy data available in various formats, and the group of GIS users is far from static, with new data formats constantly emerging. There has been some progress on standard algorithm interfaces, one each for CORBA, COM, and SQL [3]. However, the fact that three standards exist, one for each underlying technology, undermines the standardization process.

We suggest an ontological approach, one that decomposes data and algorithms into their descriptive components, independent of a data format or programming language, is superior. We have developed an ontology for an agent-based GIS by focusing on the three core components of a complete GIS: data, algorithms, and the user-posed query/results. Our ontology is based on the assumption that data and algorithms have been developed for legacy systems, and it needs to address the interoperability problem. This ontology is represented in the Resource Description Framework (RDF) encoded in the eXtensible Markup Language (XML). This ontology is used in an agent-based GIS [5] that processes user-defined queries as discussed above. The ontology serves as the basis for our agent communication language.

2. RDF AND XML

The Resource Description Framework (RDF) and the eXtensible Markup Language (XML) are both W3C recommendations for sharing information over the web. RDF allows for the capture of knowledge objects in *Resources, Properties,* and *Statements.* Relations can then be built between these objects. Others [6] have shown the usefulness of using RDF for modeling ontologies, and we have chosen this standard as a way to encode our ontology, which we now discuss.

3. COMPONENTS OF A GIS ONTOLOGY

We propose that the ontology for a GIS is fundamentally composed of three core elements: the *data* used in processing, the *algorithms* that process the data, and the *query and resultant information.* By defining a common vocabulary for these three core elements, we allow for interoperability among heterogeneous systems, as well as the potential to increase the number of tasks offloaded to automated processes

*This is work is supported under the National Imagery and Mapping Agency University Research Initiative Program.

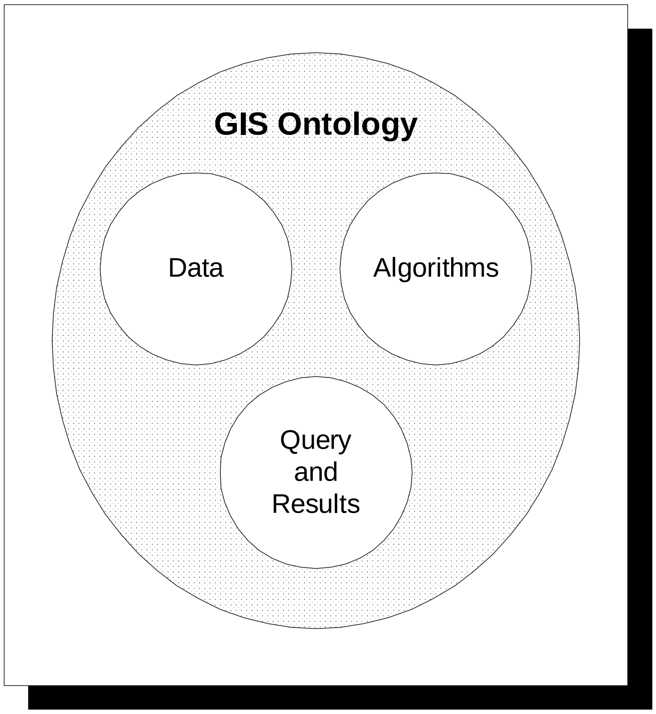

Figure 1: GIS Ontology

in the form of agents. We have created an ontology for the three fundamental areas of a GIS, a graphical view of the ontology can be seen in Figure 1.

We have found that largely, all of the information necessary to construct a GIS ontology exists in some form. In fact, much of it exists in some machine-readable form. However, the formats they have been developed in are largely heterogenous, as different organizing bodies have done the development. In the remaining sections, we discuss the three core components of our ontology, and discuss how they are used in our agent-based GIS.

3.1 Imagery and Geospatial Data

The imagery and geospatial communities have developed many data specifications that suitably describe imagery and geospatial data concepts. For example, an image in the National Imagery Transmission Format (NITF) contains metadata information that thoroughly describes the data. The same can be said for geospatial formats such as Vector Product Format (VPF) or the Spatial Data Transfer Standard (SDTS). While the aforementioned standards describe imagery and geospatial data concepts, what these standards lack are the relationships between those concepts. By taking these data specifications, and providing the necessary relationships, we submit that this can be used to form an ontology for imagery and geospatial data.

Fundamentally, all of these data types share some common characteristics. Some of these characteristics include: the **Name** of the data set, the **Type** of data set (e.g., vector, raster, or image), the spatial **Scale or Resolution** that the data represents, the geographic **Location** of the data (e.g., a point, polygon, or a rectangle representation), the **Format** of the data, the **Map Projection** of the data, any **Map Projection Parameters**, and the **Collection Date** of the data. Additionally, the data set may have other information pertaining to the creator, contact information, or other metadata.

A recent effort, the Geography Markup Language (GML) has proposed a standard for the encoding, transportation, and storage of geographic information. We submit that GML serves as a starting point for an ontology for geospatial data. This specification defines the structure and some semantics for geospatial data in XML. Similarly, NITF also is a standard used for the encoding, transportation, and storage of image information. We have implemented these standards, and slightly altered them to be suitable for use in an agent-based system. NITF is a fixed-field format, however, it uses the concept of "tags" similar to other markup languages. We have also developed an RDF implementation of the NITF standard.

3.2 Imagery and Geospatial Algorithms

Imagery and geospatial processing algorithms are fundamentally composed of: a name, a required and/or optional set of parameters, input data types, and output data types. In addition, there may be other descriptive information such as the service creator, or documentation on the service. For example, to perform image convolution, the name of the operation is "Convolve", the parameters are an input processing kernel and the service requires one image. We submit that any imagery or geospatial processing algorithm can be broken down into: the **Name** of the service, the **Parameters** required by the service, the **Number of Inputs** inputs required by the service, the **Data Type** of the inputs (e.g., image, vector), the **Number of Outputs** produced by the service, and the **Data Type** of the outputs produced by the service.

We have created an ontology for imagery and geospatial processing algorithms based on the concepts described above. Image processing service descriptions are based on the Java Advanced Imaging API (JAI). Geospatial processing descriptions are based on the OpenMap API. We have taken these two commercial packages, and identified the individual processing components by using the descriptors above. This ontology is extensible; any new algorithm may be added that can be described by our ontology components. An illustration of how the "Convolve" service would be represented in our ontology is shown in below. This example uses the RDF representation. The complete ontology for imagery and geospatial services can be found at `http://aiga.cs.gmu.edu/ontology/aiga-services.rdfs`.

```
<rdf:RDF xml:lang="en"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/TR/1999/
        PR-rdf-schema-19990303#"
    xmlns:dc="http://purl.org/dc"
    xmlns:aiga="http://aiga.cs.gmu.edu/
        ontology/aiga-services.rdfs">

    <aiga:Service>
        <aiga:name>Convolve</aiga:name>
        <aiga:creator>Sun Microsystems</aiga:creator>
        <aiga:parameter>
            <aiga:parameterType>Kernel</aiga:parameterType>
        </aiga:parameter>
        <aiga:inputDataType>Image</aiga:inputDataType>
        <aiga:numInputs>1</aiga:numInputs>
        <aiga:numOutputs>1</aiga:numOutputs>
    </aiga:Service>
</rdf:RDF>
```

3.3 Ontology as a Foundation for the ACL

According to [2] an agent communication language that allows agents to interact while hiding the details of their internal workings will result in agent communities able to tackle problems no individual agent could. Above, we described an ontology that hides such implementation details from agents. Agents are able to describe themselves using this ontology, and other agents can interpret and use the functionality of those agents. We use the described ontology as the core part of our Agent Communication Language, which we term I-XML. In this section, we describe the components of I-XML.

The **Query** section represents a question that an analyst wishes to have answered. This query is decomposed into keywords and location, the first step in the processing chain. This is the basis, or trigger, for all actions within the AIGA architecture.

The **Baseline Representation** contains information about the geographic location of the query. For example, this may include a bounding rectangle of the region of interest or a place name such as a country or city. We use GML to represent geospatial features and information.

The **Computational Steps** represent the steps necessary to answer the query. It is important to note that this is only the *necessary processing steps* and not the order in which they will be done. It is essentially a listing of the steps required to determine the resultant information, however this list has not been optimized to take advantage of any parallel processing opportunities.

The **Processing Strategy** refines the Computational Steps into a directed processing graph, which is the exact series of steps required to minimize the time required to complete the task. This is an important feature of time-sensitive systems such as imagery and geospatial systems. The Processing Strategy provides a way to represent these multiple processing chains as independent of one another, and identifies them as processing chains that could be executed in parallel if more than one processing resource is available on the network.

The notion of a Processing Strategy allows us to more efficiently utilize resources on the network and presents the opportunity for reduced processing timelines. The processing strategy tags tasks that can be done in parallel, allowing agents to move to remote locations for processing, and later synchronize results. This task distribution reduces the clock wall-time, addressing a major issue in the intelligence analysis problem domain.

The **Results** tag represents any outputs of services that may help to answer the query. As the query is executed and results are returned from agents, the Results tag will be updated with information that may include geospatial features, references to image sources, or open literature that was useful during processing.

4. ONTOLOGY AS AN INFORMATION RETRIEVAL MECHANISM

We have found that by using a well-defined ontology, we have created a mechanism that allows clients *and* agents to search for data types and agents available on the network. We have implemented Salton's vector model information retrieval algorithm [4], and we use this as the basis to match queries up with appropriate agents, who in turn make use of the algorithm to find other agents to assist in processing. The algorithm works as follows:

$$sim(a_j, q) = \frac{A_j \bullet Q}{\mid A_j \mid * Q}$$

where $sim(a_j, q)$ represents the similarity of agent j to query q, A represents a vector of agent descriptions, and Q represents a vector of the terms from query q. This formula states that the similarity of the capabilities of an agent *a(j)* to a particular query q can be calculated by taking the cosine of the angle between the vectors A (the terms of the agent description) and Q (the terms of the query).

The terms available in the repository of agent descriptions is updated each time an agent enters or leaves the network. Using these terms, each agent has the capability to calculate its relevance to specific queries posed by a user, and also search for agents *it* may require assistance from during processing.

5. SUMMARY

We have presented an ontology and ACL for an agent-based Geographic Information System. The ontology and ACL have been developed using industry standards, and relationships between those standards have been created through the use of the Resource Description Framework. The ontology and ACL have been implemented in a Java-based agent GIS using commercial packages for agent functionality, Jini as the middleware, and RDF encoded in XML to represent ontological information as well as the ACL. The ontology and ACL have been constructed in such a way that clients and agents can search for other agents using implementation independent semantics. We have demonstrated this system using over 100 GIS agents to formulate responses to open ended queries such as we have previously describes.

6. REFERENCES

[1] Y. Bishr. Overcoming the semantic and other barriers to gis interoperability. *International Journal of Geographical Information Science*, 12(4), 1998.

[2] Y. Labrou, T. Finin, and Y. Peng. Agent communication languages: the current landscape. *Intelligent Systems*, 14(2):45–52, March-April 1999.

[3] OGC. *The OpenGIS Consortium Specifications*. http://www.opengis.org.

[4] G. Salton. *The SMART Retrieval System*. Prentice Hall, Englewood Cliffs, NJ, 1971.

[5] R. Simon, J. J. Nolan, and A. K. Sood. A light-weight agent approach for collaborative multimedia systems. *To appear in: Journal on Virtual Environments and Distance Learning*, 2001.

[6] S. Staab, M. Erdmann, A. Mdche, and S. Decker. An extensible approach for modeling ontologies in RDF(S). In *Proceedings of the ECDL-2000 Workshop "Semantic Web: Models, Architectures and Management*, September 2000.

[7] P. Vittorini and P. DiFelice. A Java RMI-based application supporting interoperability in a GIS context. In *Technology of Object-Oriented Languages and Systems, 1999.* , pages 428–439, February 1999.