CHAPTER 1

INTRODUCTION

Geographic information is critical to the effective and collaborative decision making in earth-related disaster planning, crisis management and early-warning systems. The decision making in Geographic Information Systems (GIS) increasingly rely on the analyses of spatial data in map-based formats. Maps are complex structures composed of layers created from distributed heterogeneous data and computation resources belonging to the separate virtual organizations from various expert skill levels.

We propose federated Service-oriented architecture for understanding and managing the production of knowledge from the distributed observation, simulation and analysis through integrated data-views in the form of multi-layered map images.

We also address performance issues result from interoperability and extensibility requirements such as using XML-encoded common data model and Web Services' XML-based message protocol, and propose novel design approaches for structured data transfer and handling, and federator-oriented load balancing and caching.

1.1. Motivation

Geographic Information Systems (GIS) are systems for creating, storing, sharing, analyzing, manipulating and displaying spatial data and associated attributes.

The general purpose of GIS is extracting information/knowledge from the raw geo-data. The raw-data is collected from sensors, satellites or other sources and stored in databases or file systems. The data goes through the filtering and rendering services and presented to the end-users in human recognizable formats such as images, graphs, charts etc. GIS are used in a wide variety of tasks such as urban planning, resource management, emergency response planning in case of disasters, crisis management and rapid responses etc.

Over the past decade, GIS have evolved from the traditional centralized mainframe systems to desktop systems to modern collaborative distributed systems. Centralized systems provide an environment for stand-alone applications in which data sources, rendering and processing services are all tightly coupled and application specific. Therefore, they are not capable of allowing seamless interaction with the other data or processing/rendering services. On the other hand, the distributed systems are composed of geographically distributed and loosely coupled autonomous hosts that are connected through a computer network. They aim to share data and computation resources collaborating on large scale applications.

Modern collaborative GIS require data and computation resources from distributed virtual organizations to be composed based on application requirements, and accessed and queried from a single uniform access point over the refined data with interactive display tools. This requires seamless integration and interaction of data and computation resources. The resources span over organizational disciplinary and technical boundaries and use different client-server models, data archiving systems and heterogeneous message transfer protocols.

Furthermore, GIS particularly used in emergency early-warning systems like homeland security and natural disasters (earthquake, flood etc) and crisis management applications require quick responses. However, because of the characteristics of geo-data (large sized and un-evenly distributed such as distribution of human population and earthquake seismic data), time-consuming rendering processes and limited network bandwidth, the performance and responsiveness stand as the toughest challenges in distributed modern GIS.

These motivated us to research on a federated Service-oriented responsive Geographic Information System framework enabling sharing and integration of heterogeneous data and computation resources for the collaborative decision support applications requiring quick response times.

1.2. Why Federated Service Oriented Design

The proposed federated Service-oriented information system framework supports collaborative decision making over integrated data views, described in layer-structured hierarchical data provided by a federator. The users access the system as though all the data and functions come from one site. The data distribution and connection paths stay hidden and formulated as hierarchical data defined in federator's capability metadata. The users access the system through integrated data-views (maps) with the event-based interactive mapping display tools. Tools create abstract queries from the users' actions through action listeners and communicate with the system through federator.

Federation is based on federating service-oriented standard GIS Web Services capabilities metadata and their standard service interfaces about data access/query and rendering. Creating such a federated design has some advantages in data sharing, reliability, and system growth (interoperability and extensibility).

Capability is a metadata about the data and services together. It includes information about the data and corresponding operations with the attribute-based constraints and acceptable request/response formats. Compared to Web Service Description Files (WSDL), Web Services provide key low level capability but do not define an information or data architecture. These are left to domain specific capabilities metadata and data description language (GML). Capability also provides machine and human readable information that makes integration and federation of data/information easy. It also enables developing application based standard interactive re-usable client tools for data access/query and display.

1.2.1. Architectural Design Features

Federated Service-oriented GIS framework is composed of two parts. One part is consists of interoperable Service-oriented GIS components compliant with Open Geographic Standards, and other part is federator composing those components according to the application requirements by providing integrated data-views in its capability metadata.

Regarding the first part, service descriptions, standards service APIs and capability definitions are defined in standard specifications published by Open Geographic Standards. According to standards we developed two types of services. These are map-data rendering services (Web Map Services (WMS)) and data providing services (Web Feature Services (WFS)–developed by Aydin G.). In the system there are two types of data, vector data provided by WFS in the form of XML-encoded common data model (GML) and binary map-images provided by WMS. GML is a data description language which is XML encoding of the heterogeneous data. It consists of two parts, content (core data) and presentation (attribute and geometry elements). The common data model's properties enable data to be displayed, queried and easily integrated.

In summary for the first part, we basically developed a GIS framework according to the standard specifications and then, enhanced it with Web Service-based Service Oriented Architecture (SOA) principles through WS-I standards. The standard GIS Web Services' (WMS and WFS) definitions are also extended with the streaming data transfer capability by introducing topic-based publish/subscribe message oriented middleware into the system. The system uses Web Service interface as negotiation protocol and data transfers are done through publishers and subscribers on the same topic agreed on by negotiation process.

The second part is the federator which federates the standard GIS Web Services' capabilities metadata and presents a single database image to the user through application-based hierarchical data defined in its federated capability metadata. This enables unified data access/query/display from a single access point through abstract queries from event-based interactive map tools (or even from the console or command-lines). Event-based interactive map tools are generic tools enabling seamless interaction with the system through federator or any other compatible WMS.

Application-based hierarchical data is defined as integrated data-view in the federator's capability metadata. It actually defines a static workflow starting from the federator and ending at the original data sources (WFS serving GML or WMS serving map layers). The services are linked through the references defined in their capability metadata. The user's interaction with the system is carried over the integrated data views through event-based interactive map tools. Integrated data views are defined in the hierarchical data format as explained below:

Application ->Map -> Layer -> Data {GML and/or binary map images} ->Raw data (any type).

Map is application-based human recognizable integrated data display, and composed of layers. A layer is a data rendering of a single homogeneous data source. Layers are created from the structured XML-encoded common data model (GML) or binary map images (raster data).

Heterogeneous data source are integrated to the system through the mediators in the form of GML (WFS-based mediation) or binary map images (WMS-based mediation). The mediators have resource specific adaptors for request and response conversions and appropriate capability metadata describing the data and resources.

Our experiences with GIS showed that federated Service-oriented GIS-style information model can be generalized to many application areas such Chemistry and Astronomy. We call this generalized framework Application Specific Information System (ASIS) and give blueprint architecture in terms of principles and requirements. Developing such a framework requires first defining a core language (such as GML) expressing the primitives of the domain, second, key service components, service interfaces and message formats defining services interactions, and third, the capability file requirements (based on core-language) enabling inter-service communications to link the services for the federation.

1.2.2. High-performance Design Features

The high-performance design issues addressed in the proposed framework can be grouped into two. First group is regarding the extension of service descriptions and specifications of Open Geographic Standard specifications, and second group is regarding the federated design.

The first group of design issues is related to the extension and enhancements over Open Geographic Standards (OGC). We extended OGC'S online service descriptions with the streaming data transfer capabilities and called them streaming GIS Web Services. These services are the main building blocks of the federated Service-oriented GIS framework. At the service API level they provide standard functionalities and interfaces according to standards but the data payloads are transferred with the topic and publish/subscribe based messaging middleware. Each service has a publisher module and a subscriber module enabling the streaming transfer. Web

Service interfaces are used for negotiation enabling client (subscriber) and server (publisher) to agree on the IP-port and topic of the broker through which data will be streamed.

The second group of design features is regarding the federator. The federation framework inspired us developing some performance-oriented designs such as *pre-fetching*, *caching* and *parallel processing enabling* quick responses. Since the geo-data is in massive sizes, un-evenly (geo-location) distributed and variable sized, application of these techniques requires us to introduce novel approaches. These design approaches are addressed in the following paragraphs.

We introduced one-time session-based caching. In this approach, instead of caching whole data the federator caches only the recently fetched data and uses it for serving the successive requests coming from the same session. At the end of serving every-other request, federator replaces the cached data with the recently fetched data. The technique is based on sessionID transfer through the SOAP messages. This design enables application to run on any ordinary servers not having large storage capacity as in Google Servers or any other high-performance computing servers.

Parallel processing approach is developed on the proposed caching approach mentioned earlier. It is based on decomposition of un-cached data region and assigning the partitioned regions to the separate threads. Each thread runs in parallel and creates small maps correspond to their assigned partitions. The number of partitions is defined by locality information obtained by utilizing session-based caching.

Another performance-oriented design feature is the pre-fetching. It is mutually exclusive of the other design features caching and parallel processing which are applied together. Pre-fetching is explained as fetching the data before it is needed. It basically overcomes the network bandwidth problem and repeated data conversions overheads. The pre-fetching is done by federator for the performance critical data defined in the capability metadata. The pre-fetched data is kept in federator's local file system, and the requests are served from this intermediary storage.

We test the responsiveness of the system by applying these performance enhancing designs, and analyze the results by comparing the baseline tests results obtained by using naïve conventional GIS approaches (no-caching, single-threaded on demand data access/query). Our aim is turning compliance requirements into competitiveness and providing high-performance responsive geographic information systems under the interoperability and extensibility requirements.

1.3. Summary of Contributions

We developed a framework for federated Service-oriented Geographic Information Systems. Under this title, we addressed interoperability issues by integrating Web Services with Open Geographic Standards for supporting interoperability at both data, service and application levels. We also enhanced the standard GIS Web Service components with the streaming data-transfer capability by using publish/subscribe based messaging middleware architecture.

On top of the proposed service-oriented GIS data-grid, we introduced a federator enabling unified data access/query/display from a single access point through integrated data-views as a composition of distributed heterogeneous GIS data sources. Under this title, we introduced hierarchical data for integrated data-views defined in federator's capability metadata.

We have also investigated performance efficient designs regarding federator, data transfer and rendering, and done detailed benchmarking over real GIS application requiring quick response times. We compared and contrast the proposed streaming GIS Web Services with non-streaming counterparts. We have developed pre-fetching algorithm for archived data access and display, and compared it with on-demand fetching.

We have introduced novel parallel processing technique with attribute-based query partitioning for unevenly distributed variable-sized data processing. In order to do that we have proposed set of techniques such as mapping browser-based sessions to Web Services and introducing session based caching carried out between federator and event-based interactive mapping tools. We have also introduced the techniques for workload forecasting by using session-based caching. This enabled us to find out the best number of partitions for the parallel processing giving the best performance results.

Finally, we have defined the principles for generalizing federated GIS to the general science domains as Application Specific Information Systems (ASIS).

Regarding the system software contribution, we have developed streaming and non-streaming versions of Web-based GIS Map Server (WMS) with Open Geographic Standards and in Web Service principles. We also developed a federator supporting high-performance designs such as pre-fetching and parallel processing with session-based caching. We have also developed generic even-based interactive map tools for data access, query and display.

1.4. Research Issues

In order to develop federated Service-oriented GIS framework supporting event-based unified data access/query/display from a single access point we address the following research issues.

Interoperability and extensibility: We first investigate the adoption of Open Geographic Standards to GIS to create an *interoperable geographic* information system with standard data models, service description and service API, and service capabilities metadata. Second, we apply Web-Service principles to develop Service-oriented Architecture for GIS data-grid.

Federation: Another research issue will be investigating a framework for capability file-based federation of GIS data-grid services enabling unified data access/query/display through event-based interactive tools over integrated data-views. Federation is done by capability metadata federation of proposed GIS Web Services by the federator.

We investigate how to make capability federation to develop application-based hierarchical data definitions in federated capability file. We first define GIS Web Services (extension of OGC standards) and their service API allowing inter-service communication through capability metadata exchange. We also investigated the standard event-based interactive query and display tools for the standard GIS data services enabling unified data access/query/display over integrated data-views.

We also investigate the principles for generalizing the proposed federated GIS system for general science domains such as Chemistry and Astronomy in terms of components and framework requirements. This includes defining required services types, generalizing the service interfaces and message formats, defining capability format meeting the requirements for inter-service communications to link and cascade the services for the federation.

Performance and Responsiveness: Interoperability requirements bring up some compliance costs. These are due to using XML-encoded common data model and GIS Web Services (SOAP over HTTP protocol for message exchange).

We first investigated the performance efficient designs for XML structured data transfer and processing (parsing and rendering). Second, we research on federator oriented design features to support high-performance for Geographic Information Systems requiring quick response times.

1.5. Organization of Dissertation

[COMING LATER…]