Thesis Proposal

Web Services as Federated Data and Information Filters to Support Distributed Access, Query, and Transformation

Author: Ahmet Sayar
Indiana University
Department of Computer Science

Advisor: Prof. Geoffrey Fox
Indiana University
Department of Computer Science
Department of Physics
School of Informatics

CONTENTS

Abbreviated Terms 3

ABSTRACT 5

1. Introduction 6

2. Literature Review 10
  2.1. Matching ASIS Components to the Components in Major Science Domains .. 11
  2.2. Scientific Data/Information Grid 12
  2.3. Data Access and Federation 13

3. Preliminary Results and Discussions 14
  3.1. Capabilities File as Metadata in Filter Services 16
  3.2 An Example of the Capabilities Federation in GIS Domain 18
  3.3. Implications of the Compasibility Nature of the Filters through Capabilities Federation 21

4. Summary and Conclusions 22
  4.1. Summary of Research Issues 22
  4.2. Expected Contributions 23

5. Milestones 24

REFERENCES 25

Abbreviated Terms

ADQL	Astronomy distributed query language
ASFS	Application Specific Feature Service
ASIS	Application Specific Information Systems
ASL	Application Specific Language
ASVS	Application Specific Visualization Service
BPEL	Business Process Execution Language
GCL	Community Grids Lab
CML	Chemistry Markup Language
CRS	Coordinate Reference System
DIKW	Data-Information-Knowledge-Wisdom
DMF	Data Migration Facility
FITS	Flexible Image Transfer
FTHPIS	Fault Tolerant High Performance Information Service
GEON	Geophysics Network Project
GIF	Graphics Interchange Format
GIS	Geographic Information Systems
GML	Geographic Markup Language
HTTP	Hyper Text Markup Language
IVOA	International Virtual Observatory Alliance
IDS	Interactive Decision Support
JPEG	Joint Photographic Experts Group

LEAD	Linked Environment for Atmospheric Discovery Project
MCAT	Metadata Catalog Service for SRB
MCS	Metadata Catalog Service for OGSA-DAI
MSS	Mass Storage System
NVO	National Virtual Observatory
OGC	Open Geospatial Consortium
OGSA-DAI	Open Grid Service Architecture – Data Access and Integration
PNG	Portable Network Graphics
SOAP	Simple Object Access Protocol
SRB	Storage Resource Broker
SVG	Scalable Vector Graphics
SRS	Spatial Reference System
UCD	Unified Contend Descriptor
VOResource	Virtual Observatory Resource
VOTable	Virtual Observatory Tables
WFS	Web Feature Service
WMS	Web Map Server
XML	Extensible Markup Language

ABSTRACT

Statement of the research problem: Geographically distributed heterogeneous data are administrated by different organizations, are provided in a variety of formats, and are stored using a number of technologies (such as databases, file systems, resource brokers, and so on). Accessing and integrating different data from different providers using different heterogeneous technologies is a difficult task. Furthermore, multiple data sources not only need to be integrated but also transformed into comprehensible representations through cascaded inter-service communications. The attributes of the data/information which are building these comprehensible representations should be interactively accessed and queried.

Proposed Solution: Other systems such as the Storage Resource Broker and OGSA-DAI have examined various general approaches to data federation. However, we propose an alternative solution based on what we call "capability" federation of Web services. In our approach, each service is described by generic and domain specific metadata descriptions that can be queried through Web Service invocations. Going beyond the enablement of service discovery, this approach enables at least three important things. First, services of the same type that provide a subset of the request can be combined into a "super-service" that spans the query space and has the aggregate functionality of its member services. Second, the capability metadata can be used to determine how to combine services into filter chains with interconnected input-output ports. Third (and building on the previous two), capabilities of "super-services" can be broken into smaller, self-contained capabilities that can be associated with specific services. This enables performance gains through load-balancing.

Test and Experiments: Initially, we will implement our proposed solution for a single domain area, Geographic Information Systems (GIS). We will implement the Open Geospatial Consortium's Web Map Service (WMS) and will integrate it with GIS and other services for data access, information management, and scientific plotting. This will involve cooperation with other lab research projects in GIS and will be integrated with geophysical applications from the ServoGrid project. The GIS services will be described with capability metadata that will test our proposed framework. From this work we will demonstrate the basic functionality of our system and our capability-based approach. We will demonstrate WMS federation and Map-Plotting service integration. Based on this work, we will design experiments that will illustrate capability "meta-querying" and service aggregation and federation into "super-services" described above. We will also compare and contrast capability federation to more traditional data federation.

Expected Contributions: Based on our tests of GIS-based services, we will analyze other domains (astronomy and chemistry/chemical informatics) for their suitability to our approach. From this, we will develop an approach for designing abstract service specifications that can be federated using our capability approach. We will distinguish when such approaches are applicable and when they are not.

1. Introduction

In science applications, scientists need to access remote data provided by different administrative domains. Data might be raw data coming from real-time sensors or instruments, or it may be information and knowledge which are processed and interpreted by a chain of services defined statically or via workflow systems. Furthermore, in order for the data to be useful for the decision making, it is transferred into comprehensible representations through cascaded inter-service communications, and interactively queried and accessed in the same way.

Scientific data is complex and heterogeneous, even if they come from the same domain; different vendors provide data in different formats. In that case, federating and searching the data requires a lot of extra application and data specific implementation work. In order to solve these heterogeneity and complexity problems (at least in a specific domain), some standard bodies (such as Open Geospatial Consortium (OGC) [2] in the Geographic Information Systems (GIS) domain and International Virtual Observatory Alliance (IVOA) [3] in the Astronomy domain) have created standards for the online services and data models.

In our analysis of these systems, we find the following. In order to provide interoperability across the platforms, operating systems, and programming languages, standard bodies define a) required services and their interfaces, b) functionalities each service provides, c) message formats and supported communication protocols, and d) data annotation and metadata formats and concepts. Standard bodies for science domains mostly do not define specifications for the data federation but their definition of data model enables federation indirectly. Among these data models are Geographic Markup Language (GML) [5] for GIS and Virtual Observatory Tables (VOTable) [17] for IVOA.

We propose an architecture for *capability federation* that will enable inter-service communications for data/information access and querying over comprehensible representations. Other systems such as SRB [25, 26] and OGSA-DAI [23, 24] examine various general approaches to data federations. In our approach, main actor is the XML based *capability document*. Each Web Service is capability enabled. Going beyond the enablement of service discovery, this approach enables at least three important things. First, services of the same type that provide a subset of the request can be combined into a "super-service" that spans the query space and has the aggregate functionality of its member services. Second, the capability metadata can be used to determine how to combine services into filter chains with interconnected input-output ports. Third (and building on the previous two), capabilities of "*super-services*" can be broken into smaller, self-contained capabilities that can be associated with specific services. This enables performance gains through load-balancing.

Information federation is constructed on top of the data/data-federation services. There is no clear separation line between data and information; it changes from application to application. The process for transforming raw data into human comprehensible forms is referred to as the Data-Information-Knowledge-Wisdom (DIKW) hierarchy [30]. Information is the meaning that is currently assigned to data by means of the conventions applied to these data. Knowledge is an organized, integrated collection of facts and generalizations [4]. Wisdom is super extraction from information and knowledge. In order for the scientific data to be useful, information has to be extracted and converted to wisdom decisions or at least knowledge. However, because of the complexity, heterogeneity and multidisciplinary nature of the scientific data, it is difficult to extract knowledge and draw wisdom decisions from them without complex chained processes. In our approach to the problem solution, each process in the chain is wrapped as a Web Service [34] and called as "*Filter*". See the Figure 1 for the sample scenario. The user lists the available layers by using Interactive Decision Support (IDS) [35, 36] tools and selects data A and data E to access, display and query interactively. In this scenario, on the way to access the data A, there will be three Filters visited. Even if the first Filter says that it provides data A in its capabilities file, it does not mean that it provides data from its Databases or file systems. Provided data and corresponding Filter addresses are defined in the Filters' capabilities files. If the Filter is capable of communicating and obtaining data from other Filters, and updates (or aggregates) its capability metadata with these data (after capability files exchange), then it can claim that it serves those data.

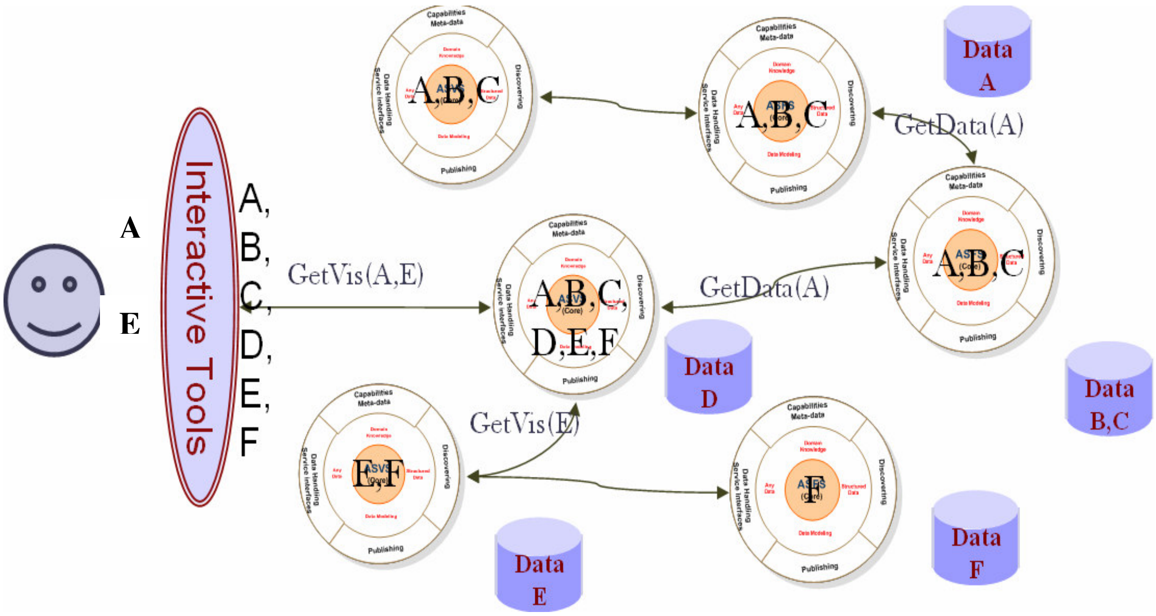


Figure 1: Successive requests are done, user is not involved. These request chains are created based on the filters' capabilities that published before

Filter Services are information sources (not data – see previous paragraph) that enable distributed data/information access, querying and transformation through their predictable input/output interfaces defined by *capability documents*. A capability document is an XML based document that provides metadata about the filters and their functionalities. A Filter Service inputs DIKW from other Grids or Services and outputs DIKW – perhaps converting data to information, comprehensible data representations etc.

Services in the science domains or in any other domains take some parameters as inputs and after processing/transforming return something as outputs. For the sake of generality we call these services filters after adding some specific functionalities, service ports and metadata capabilities. Raw data might go through several filtering operations that transform and analyze it before the scientists and end users see it. Users can just understand and interpret the data converted into the comprehensible formats. Basic filter applications typically include simple transformation operations and predictable data extraction operations (which extract out known parts of the data stream). Filters are typically grouped into chains of dependent operations that produce specialized data products through a process commonly called 'workflow'. Filters are basically Web Services. So, they are chainable together to create more complex services by using any third party workflow machines and languages such as Business Process Execution Language (BPEL) [18]. In our initial proposal, we use static chaining of filters instead of using any workflow systems. We are not interested in the problem of workflow enactment (typically performed by enactment engines such as Taverna [9] or BPEL implementations), but are instead concerned with the problem of automatically determining if two service filters can be linked in a filter chain link.

Based on our analysis of OGC GIS and IVOA standards (and others?), we have identified a number of generic system components enabling federated data and information filtering to support distributed access, querying and transformation. All the components are Filter Services in Web Services. Being a Web Service enables Filter Services to publish their interfaces, locate each other and chain together easily. We call our proposed system ASIS (Application Specific Information System). In ASIS, Filter Services are grouped into two: Application Specific Feature Service (ASFS) and Application Specific Visualization Service (ASVS). Communication is fulfilled with two main elements. These are Application Specific Language (ASL) and capabilities document of the communicating pairs. ASL is a common data model structured in XML. XML based hierarchical data model enables common language and communication across operation system and platforms to exchange and federate information. In order to make the proposed architecture feasible, all the data should be converted to ASL through an adapter deployed database end Filter Service such as Web Feature Service (WFS) [6] in GIS domain. Filter Services provide data in consistent formats and define these formats and the ways to access them in their capability documents.

ASIS is an approach for designing capability based metadata aggregation and defining abstract service specification interfaces for fulfilling basic communication requirements for accessing and querying distributed data in transformed and comprehensible formats. By the aggregation we mean federation at the capabilities level. If a Filter Service can communication with another then it can provide the others data too. So in order to express that it should update its capabilities document by aggregating the capabilities files. This aggregation can be done in a specific domain in accordance with the capabilities schema. Inner implementation of the transformation in the Filter Services might possibly change from domain to domain, but the general architectural requirements and service entities and interfaces are the same. We define the architectural requirements for the general science domains and create a complete system for the GIS (ASIS).

In order to test the proposed system and do some experiments we created the Interactive Decision Support (IDS) [1] [19] tools for the geosciences applications in the ServoGrid [8] project. We will use IDS tools for investigating the interesting general problems of information transformation and service federation. We do our test (involving cooperation with other lab research projects in GIS such as HPSearch [28] and FTHPIS [29]) on geophysics applications from the ServoGrid project. The decision support user needs human readable outputs such as images and animations and statistical information such as plotting, graph, and pie charts. When using IDS tools, users don't need to know data and application specific parameters and query settings for querying the heterogeneous data providers. They just use the tools to query and get the information and knowledge in comprehensible formats from the system to make wisdom decisions. We also created movie and animation tools for querying and creating animation on time series data for the GIS domain [1, 35].

We propose some quality of services for Filter Services. These are dynamic filter interaction to upgrade their integrated cascaded capabilities metadata files, and streaming connection between all the services that provide on the fly archiving, high-performance transport and possibly fault tolerance.

As methodology, we first create ASIS for the GIS domain, and then create client/end-user tools to enable usage of the proposed architecture. After proving that it works for GIS, we extend and define the general architecture requirements for the other domains, defining the challenges, and general characteristics and/or requirements for ASIS. Based on our tests of GIS-based services, we analyze other science domains for their suitability to our approach.

In Chapter 2, we present literature survey about first, how strongly ASIS architecture matches to major science domains such as GIS, Astronomy and Chemistry, second, well known scientific data-grid projects such as LEAD (Linked Environment for Atmospheric Discovery) [20], Geophysics Network (GEON) [21, 22], and third, major data access and federation projects in Grid environment [15, 16]. Chapter 0 consists of conclusions. It covers research issues and expected contributions in its subsections. Chapter 5 is milestones. We list what we have done and what we will be doing in order to solve our research problem.

---

2. Literature Review

This section is composed of three parts. One is regarding possible science domains to which ASIS is applicable. It gives a general idea about how strongly ASIS fits to different major science domains. The following subsection is about some related well known data-grid projects. It gives the idea about the general picture of what we are trying to achieve and what the ultimate goal in the scientific data-grid. The last subsection is about the data access and federation in Grid. It examines two well known data-grid projects. These are Open Grid Service Architecture – Data Access and Integration (OGSA-DAI), and Storage Resource Broker (SRB).

GIS is our motivating domain. GIS is a system for creating and managing spatial data and associated attributes. OGC is a standard body for the GIS domain. It defines publicly available online services and data model specifications. Its goal is to make geographic information and services neutral and available across any network, application, or platform. The most important OGC specifications are Web Map Service (WMS), Web Feature Service (WFS) and Geographic Markup Language (GML). We implemented OGC compatible WFS *(implemented by Aydin, G.)* and WMS [7] for our proposed ASIS architecture. We also use GML (we call it ASL in the ASIS) as common data representation for our geographic data such as earthquake seismic data and state boundary lines data.

WFS provides standard interfaces for data manipulation operations of geographic features such as create, delete and update. Data provided by WFS is geographic features and in the form of XML-based GML. GML represents vector data with geometry elements such as lines, polygons and points, and non-geometry elements and attributes defined based on its schema.

WMS is a map server producing maps of spatially referenced data and providing them in the form of images and some other comprehensible representation formats such as SVG, PNG, and text files. They define possible output formats in their *capability documents*. A WMS is usually invoked by a client application that provides the user with interactive controls. Client application first connects to a WMS and request its capabilities file and dynamically update its user interface based on the capabilities file. Depending on the requests WMS might need to interact with other WMS (cascaded-distributed inter service communication), WFS and information services. The detailed architecture will be explained later.

The proposed architecture is based on Web Services. The Web Service technique overcomes the shortcomings of the traditional Distributed Object technique and provides the interoperable capability of cross-platform and cross-language in distributed net environments. The Web Service architecture establishes standard interconnection rules between services and information clients that nicely support the dynamic integration of data, which is the key to creating federated information management architecture. Wrapping Filter Services such as ASFS and ASVS from different vendors as Web

Services enable them to be integrated into federated information management systems easily.

2.1. Matching ASIS Components to the Components in Major Science Domains

- OGC GIS, Chemistry and Astronomy

All these science domains have some common characteristics and this motivates scientist to work in the same or similar directions. These characteristics/challenges can be summarized below;

- Distributed nature of scientific data
- Proprietary data formats, and service methodologies
- Annotation and metadata formats and boundaries between data and metadata
- Lacking of interoperable services
- Federating/assembling data from distributed sources
- Format conversions
- Amount of resources for geo-processing
- Modular extensible visualization tools

Table 1: Matching ASIS components to well known science domains

ASIS Sci-Domains	ASL	Filters		Metadata
		ASFS	ASVS	
GIS	GML	WFS	WMS	capability.xml schema
Astronomy	VOTable, FITS	SkyNode	VOPlot TopCat	VOResource
Chemistry	CML	None	NO standard JChemPaint	None

For a simple scientific data/information grid, we need data, a data/information provider, and a data/information displayer (See Figure 2). In our proposed prototype ASIS architecture, we represent data in ASL, the data/information providers are ASFS and ASVS and the comprehensible data displayer is the client application with interactive decision support tools. After our long literature survey, we saw that other main science domains also have similar online services serve for the same/similar purposes (see Table 1). Not all the domains have specifications for all the services, data models and metadata definitions in their field. For example in Chemistry, specifications for distributed scientific applications are very immature, moreover, they have just data model specification (CML in Table 1) which is not widely accepted and used. Regarding Astronomy, FITS (Flexible Image Transfer), Images and VOTable are the data models. SkyNodes is a Database server with an ADQL (Astronomy distributed query language) based SOAP interface returning VOTable based results. VOPlot and TopCat are the services to visualize the astronomy data in the forma of VOTable, FITS and images. VOResource and UCD are the metadata definition and standards for the service descriptions.

2.2. Scientific Data/Information Grid

Here we talk about some major data-grid applications for various major science domains such as LEAD, GEON, MyGrid [12, 13], and National Virtual Observatory (NVO) [14]. Please see Figure 2 for the simplest case of their relations to our proposed architecture, ASIS. These major data-grid applications use projects mentioned in Chapter 2.3 for achieving the data access and integration.

We will not go into detail and explain all these sample projects but rather will give a summary of their general characteristics from our proposed architecture's point of view. All application grids that we surveyed have similar features. All use Web Service architecture. All have service families: data services, application execution services, information services, and workflow. We also use similar features in our application grid. For that purposes, we enhanced ASIS architecture by integrating it with other lab projects such as FTHPIS (for Information Services) and HPSearch (for workflow). These services are located in the cloud of "Data Grid Service" in Figure 2. But our main focus is based on capability federation of Filter Web Services.

LEAD seeks to provide on-demand weather forecasting in meteorology domain. They have myLead concept in which each user is given a file space, and context and session parameters through the portal. GEON provides ontology enabled applications mostly based on data registration, discovery, manipulation and display in the GIS domain. They also have myGEON concept functioning similarly as in the LEAD, and they have data display tools in a portal implemented by GridSphere. MyGrid is basically a combination of Grid, Web Services, and semantic web efforts in the bioinformatics domain. MyGrid supports semantic information for the data and services in the grid by using ontology services based on DAML+OIL (DARPA Agent Markup Language + Ontology Inference Layer) reasoner. NVO is a data-grid project in the Astronomy domain. NVO is compatible with the IVOA standards. IVOA is a standard body for Astronomy like OGC for GIS. IVOA define some online services and data models for the Astronomy such as VOTable and FITS for the data models and VOResource for the metadata definitions.

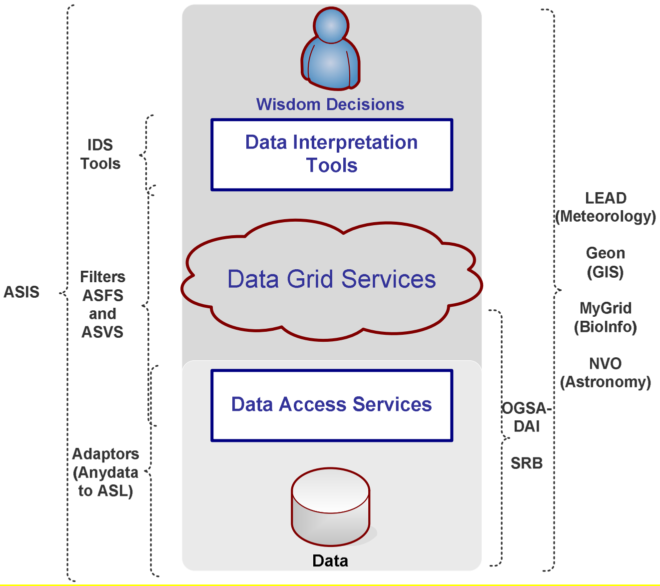


Figure 2: Related projects and where we are in the related research world.

2.3. Data Access and Federation

The related works at the data access and federation level (See Figure 2) are SRB and OGSA-DAI. SRB is much more like a data grid middleware that provides a storage repository abstraction for transparent access to multiple types of storage resources, while OGSA-DAI puts more emphasis on data retrieval in Grid environment. We are closer to SRB in capability managing data integration than OGSA-DAI with some differences.

Our differences with the OGSA-DAI are very clear as summarized in this paragraph briefly. OGSA-DAI emphasizes the Database layer, whereas we are tackling the application specific DIKW to describe the hierarchy of Data-Information-Knowledge-Wisdom that we are attempting to support. OGSA-DAI can be leveraged in ASIS. OGSA-DAI enables access to heterogeneous data via common interfaces on the Grid. The catalog service is MCS (Metadata Catalog Service) [10, 11]. Sample projects inspired from OGSA-DAI are LEAD (myLEAD) and MyGrid.

The SRB is implemented as a federated client server system, with each server managing/brokering a set of storage resources. Storage resources include digital libraries, Mass Storage System (MSS), UniTree, Data Migration Facility (DMF) and file systems. SRB is consists of three components, MCAT services, SRB servers to access to storage repositories and SRB clients, connected to each other via a network. SRB provides uniform access to distributed heterogeneous data resources by attributes. It uses MCAT (Metadata Catalog Service) [27] to enable attribute based querying and data access. SRB can be leveraged in ASIS. BIRN and NVO are sample projects using SRB.

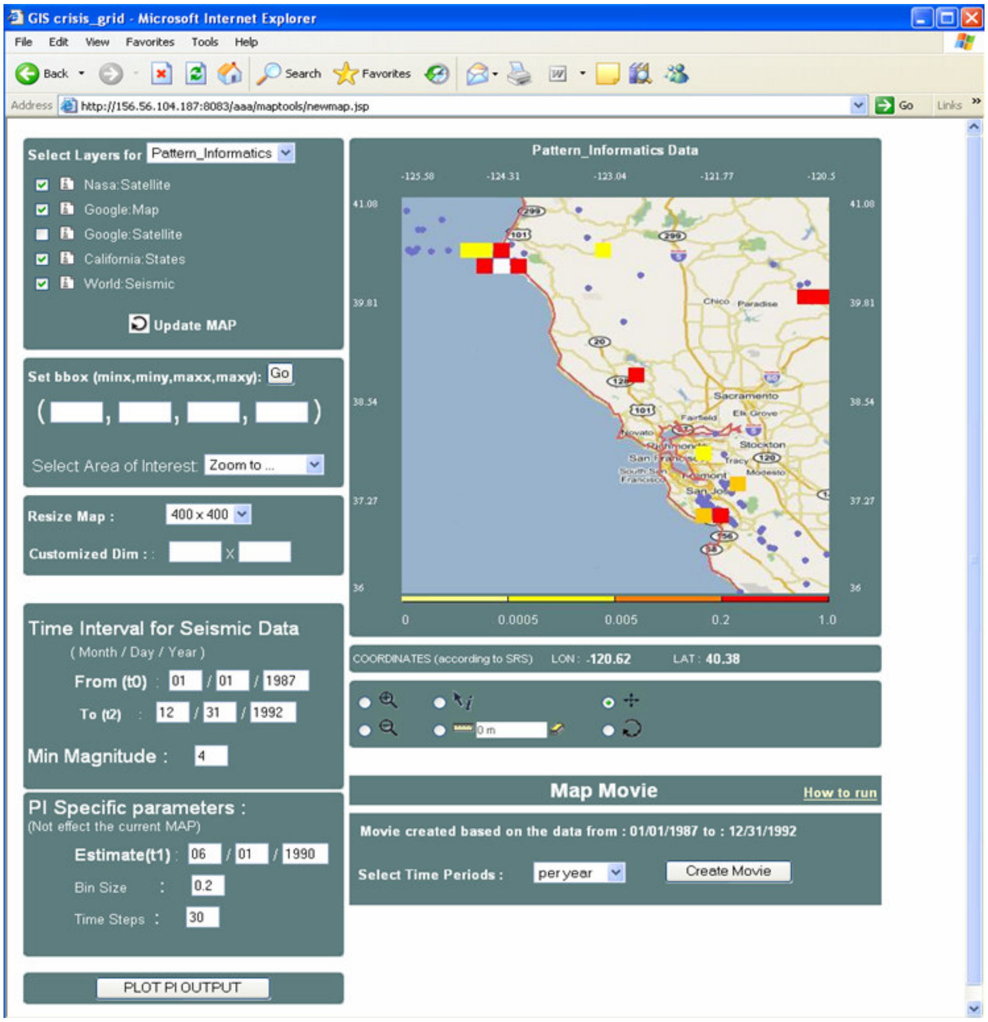
What we are doing different from SRB: In order to integrate any data to the system, we convert it to ASL. ASL is an XML based common data format. Using XML based common data provides some advantages. XML based hierarchical data model enables common language and communication across operating system and platforms to exchange and federate information. In order to make this conversion, we use adaptors in the end-filter services deployed at the service invocation points of Databases. We don't need standalone catalog services such as MCAT in SRB, instead, each component (Filter Service) keeps and handles its metadata in its local file system as an XML based capabilities file. Filter Services can update and aggregate their capabilities file through their capability exchange ports. In contrast to *SRB servers*, we have end-filter services. They have adaptors for converting any data to ASL. Our main motivation is on XML based common data model transformation through the Filter Web Services to create comprehensible and more informatory data/information for the decision makers in science domains.

The MCAT server handles requests from the SRB servers. These requests include information queries as well as instructions for metadata creation and update [31]. MCAT is implemented in tight integration with other components of SRB and is used to control data access and consistency as well as to query metadata. MCAT can not be used as standalone component. In addition, MCAT stores both logical metadata and physical metadata that characterizes file properties as well as attributes that describe resources, users and methods. By contrast, MCS is a component in a layered and composable Grid architecture [11]. Since we use predefined capabilities and common data model (fixed schema for metadata and data model) we do not necessarily need the catalog services such as MCS and MCAT, and metadata creation and update. This property of our system makes our proposed architecture easy to use and understand, but it requires additional internal and domain specific implementation extension or changes when the application users need to use the same architecture for some other domains. We have two fundamental elements in the proposed architecture. These are metadata in capability document and common data model in ASL. They change from domain to domain and should be well defined in an XML schema file to enable capabilities federation of Web Services and accessing and querying the data through the filter-chain.

3. Preliminary Results and Discussions

We basically propose middleware architecture for scientific applications that have the requirements to access, query and process many data and information sets. We focus on the capability metadata and capabilities federation through inter-service communications to create the proposed middleware architecture enabling distributed query access and transportation of the data and metadata. Filters are basically data providers. Each filter handles its own data access requirements and database access jobs. Each filter has basically two ports for two independent channels. These are the port for the metadata communications and federations, and port for data transportations. We describe the requirements for metadata services in the context of Grid and the available Grid Services (Web Service based).

In order to test usability and performance of our proposed system and use it in the ServoGrid projects we have developed a portlet-based IDS tools interface A sample interface is shown in Figure 3. Several capabilities are implemented for the user to access and display geospatial data. Tools enable the user to zoom in, zoom out, measure the distance between two points on the map for different coordinate reference systems, get further information based on the attributes of the features on the map, and drag and drop the map to display different bounding boxes. Users can also request maps for the area of interest by selecting predefined options, clicking the drop-down list. The user interface also allows the user to change the map sizes from the drop-down lists or enable them to give specific dimensions. Zoom-in and zoom-out features let the user change the bounding box values to display the map in more or less details.



Address: http://156.56.104.187:8083/aaa/maptools/newmap.jsp

Select Layers for: Pattern_Informatics

☑ Nasa:Satellite
☑ Google:Map
☐ Google:Satellite
☑ California:States
☑ World:Seismic

Set bbox (minx,miny,maxx,maxy): (, , ,)

Select Area of Interest: Zoom to ...

Resize Map : 400 x 400

Customized Dim : : x

Time Interval for Seismic Data
(Month / Day / Year)
From (t0) : 01 / 01 / 1987
To (t2) : 12 / 31 / 1992
Min Magnitude : 4

PI Specific parameters :
(Not effect the current MAP)
Estimate(t1) : 06 / 01 / 1990
Bin Size : 0.2
Time Steps : 30

COORDINATES (according to SRS) LON : -120.62 LAT : 40.38

○ ○ ●
○ ○ 0 m ○

Map Movie
Movie created based on the data from : 01/01/1987 to : 12/31/1992
Select Time Periods : per year

Figure 3: A snapshot from Interactive Decision Support Tools Interface. Sample snapshot is from Pattern Informatics Demo.

Our implementation of IDS tools consist of application independent client tools for the OGC compatible GIS services. They can support more than one geophysics application at the same time with some application based extensions. Each geophysics application is bound to a set of layers. These bindings are defined in a structured XML formatted properties file. Users navigate over the applications by selecting a set of layers from the dropdown list. A set of layers in the dropdown list is created according to communicated

WMS (proposed ASVS). Binding properties are updated based on the set of supported layers of the communicated ASVS.

In the initial implementation, Filter Services are stateless services. Since each Filter Service wrapped as Web Service and interactions are done through the SOAP over HTTP, at each interaction, all the state information and environment parameters are renewed. In the future, we plan to add caching and load balancing capabilities to the filters, in order to make them partially state-full at the application level. If the same client wants the same data in the same session, instead of redundant repeating jobs, filters do read from the cache and send to the client in a faster and fault tolerant way (for more details see Chapter 3.3).

In addition to creating static maps, IDS tools have the ability to display time series data as movies or animations (see a sample snapshot from a demo Figure 6). Movies or animations are created by getting set of successive static images (movie frames) and playing them successively. Each frame represents a static map that corresponds to a time frame defined in the requests. We have created movies and animation for Pattern Informatics and Virtual California projects in ServoGrid [1].

In the following subsections (3.1, 3.2, 3.3) and in my thesis, we focus on the federation of capability metadata of the web services as filters in order to provide high performance distributed data access, query and transformation. In order to do that we first define metadata structure enabling filter federation and filter architecture enabling inter-service communications. We test our system and obtain performance results by using IDS tools with ServoGrid applications.

3.1. Capabilities File as Metadata in Filter Services

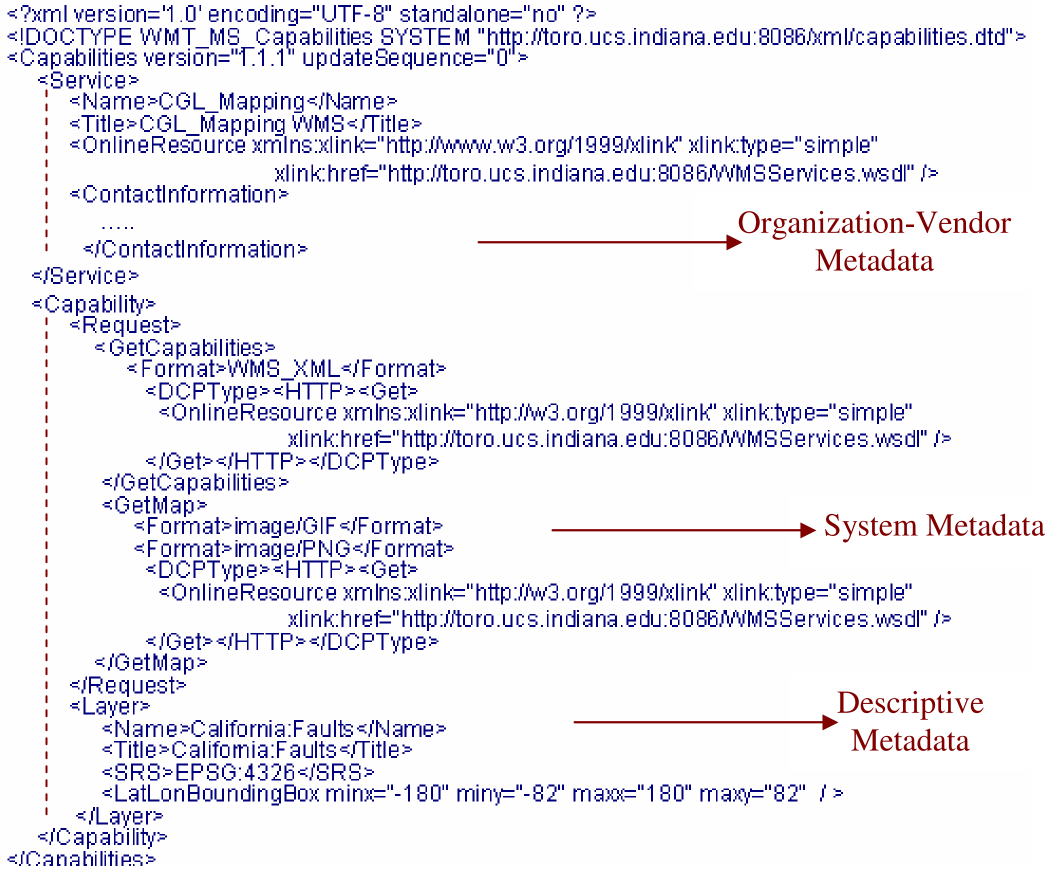
Metadata is very important for increasing the performance and efficiency of the data oriented scientific applications. Metadata is the information that describes data and allows scientists to record information about the creation, transformation, meaning and quality of data items as well as allows querying the data items based on the descriptive attributed in the metadata [11]. The boundary between data and metadata is to some extend arbitrary and may vary during a data object's lifetime. For example, for some users a table of astronomic objects derived from images is primary data, while for others it is metadata that indexes the primary or calibrated data [37].

There are various types of metadata, some metadata relate to the physical characteristics of data objects, some relate to the replication information, and some relate to the contents of the data items. In summary, metadata is grouped into 3 different categories.

- System Metadata
- Descriptive Metadata
    - Domain Specific Metadata
    - Domain-Independent Metadata
- Organization-Vendor Metadata

System metadata is mostly about the interfaces information, available service invocation points, location and access control information and protocols and binding information. Descriptive metadata describes the contents of the entire data collections and the individual data items. Descriptive metadata is grouped into two. These are domain specific metadata and domain independent metadata. Domain independent metadata consists of metadata attributes that apply to data items regardless of the application domain. Domain specific metadata attributes are often defined by metadata ontologies developed by the application communities. For example in GIS communities “bounding box” and “spatial reference system (SRS)” are the samples of the common set of terms and metrics used and agreed on by the GIS community to define any geographic data.

In Figure 4 we illustrate a sample brief capability file used as metadata in OGC compatible GIS systems for WMS. As we listed above it consists of the three different information sets. These are organization or vendor metadata, system metadata and descriptive metadata. For the capabilities federation we will be working on descriptive metadata. The others will be static after application runs but descriptive metadata will be dynamically updated depending on the change in the data availability. Based on this architecture, Filters will be updating their capability metadata without human intervention. They do that through proposed inter-service communication ports enabling capability metadata exchange between the Filters.



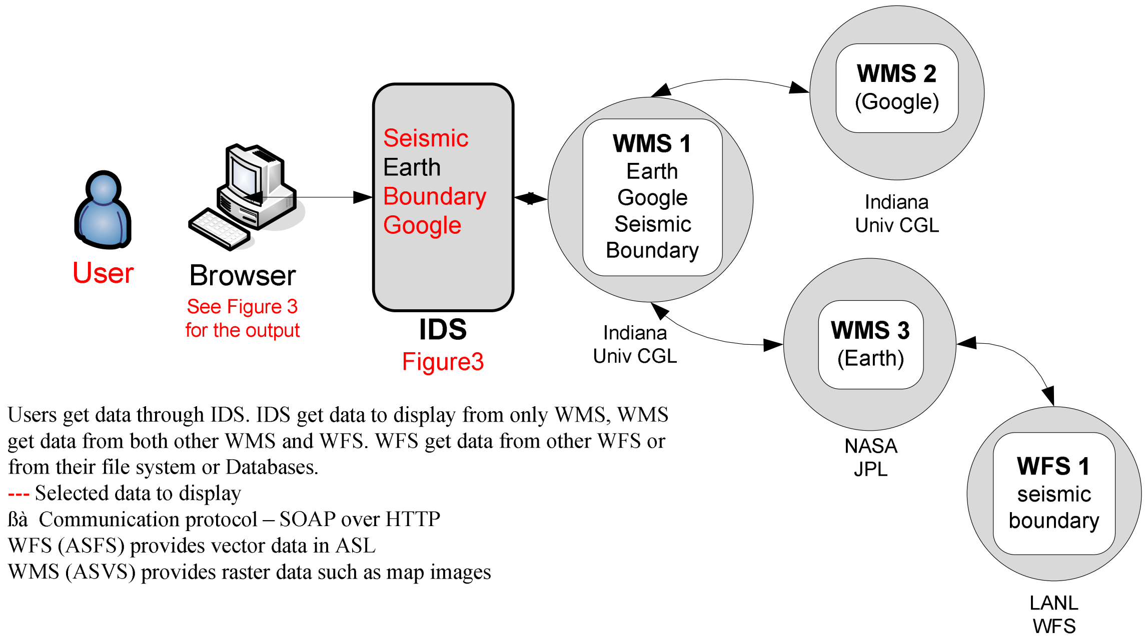
```
<?xml version='1.0' encoding="UTF-8" standalone="no" ?>
<!DOCTYPE WMT_MS_Capabilities SYSTEM "http://toro.ucs.indiana.edu:8086/xml/capabilities.dtd">
<Capabilities version="1.1.1" updateSequence="0">
   <Service>
      <Name>CGL_Mapping</Name>
      <Title>CGL_Mapping WMS</Title>
      <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
                       xlink:href="http://toro.ucs.indiana.edu:8086/WMSServices.wsdl" />
      <ContactInformation>
         .....
       </ContactInformation>
   </Service>
   <Capability>
      <Request>
        <GetCapabilities>
            <Format>WMS_XML</Format>
              <DCPType><HTTP><Get>
                <OnlineResource xmlns:xlink="http://w3.org/1999/xlink" xlink:type="simple"
                       xlink:href="http://toro.ucs.indiana.edu:8086/WMSServices.wsdl" />
              </Get></HTTP></DCPType>
         </GetCapabilities>
         <GetMap>
             <Format>image/GIF</Format>
             <Format>image/PNG</Format>
              <DCPType><HTTP><Get>
                <OnlineResource xmlns:xlink="http://w3.org/1999/xlink" xlink:type="simple"
                       xlink:href="http://toro.ucs.indiana.edu:8086/WMSServices.wsdl" />
              </Get></HTTP></DCPType>
        </GetMap>
      </Request>
      <Layer>
         <Name>California:Faults</Name>
         <Title>California:Faults</Title>
         <SRS>EPSG:4326</SRS>
         <LatLonBoundingBox minx="-180" miny="-82" maxx="180" maxy="82"  / >
       </Layer>
   </Capability>
</Capabilities>
```

Figure 4: Sample capability file used as metadata in Filter Services (OGC compatible GIS Domain).

In the proposed architecture, metadata is distributed and each filter has its own metadata about itself, its services and data it provides. Metadata is easy to read and interpret by the machines and human beings. They are in the form of capability listings. Through the filtering and capability metadata, every data provider keeps and handles its own data and metadata, and serves them upon requests via separate service interfaces. These are "*getCapability*" and "*getMap*" for the WMS case in GIS domain. WMS is considered as Filter and called ASVS in our proposed general architecture.

3.2 An Example of the Capabilities Federation in GIS Domain

According to our definitions for the proposed architecture, in the GIS domain ASVS is Web Map Server (WMS). Here we give a concrete example about how to federate capabilities metadata through cascading WMS. WMS basically provide data in the layer format. When we talk about the federation of WMS we mean federating data in the form of layers. A layer is said to be federated if it is obtained from an originating server and then included in the service metadata of a different server. The second server may simply offer an additional access point for the layer, or may add value offering additional output formats or reprojection to other coordinate reference system. Through the capabilities federation WMS obtains a layer data from another WMS, and serve that data to other clients as its own.



Users get data through IDS. IDS get data to display from only WMS, WMS get data from both other WMS and WFS. WFS get data from other WFS or from their file system or Databases.
--- Selected data to display
ßà Communication protocol – SOAP over HTTP
WFS (ASFS) provides vector data in ASL
WMS (ASVS) provides raster data such as map images

Figure 5: Sample Capabilities federation scenario for the Figure 3

In order to explain what we mean by the capabilities federation and why we need to do that, we give the sample scenario in Figure 5. Since all the servers are implemented as Web Services based filters providing standard filter interfaces, client application on IDS can also make its requests to other Filters shown in the Figure. Each filter publishes its metadata through the capabilities file in the static mode (set before run-time). You can see the schema of the OGC compatible WMS at [7]. We do not care the vendor and service part of the capability metadata for the capability federation. The most important part of the capability metadata is the descriptive part. Below you see the related parts of the capability metadata of the WMS1 filter in the Figure 5. This specific part of WMS 1 capability metadata (for the google map data) shows that Google map is going to be obtained from "*href=http://Filter_WebService_Address/*" in the form of "image/gif" and in the bounding box defined in "*EX_GeographicBoundingBox*" sub element.

```
<Layer cascaded="1">
        <Name>Google </Name>
        <Title>Google Map</Title>
        <EX_GeographicBoundingBox>
                <westBoundLongitude>-180</westBoundLongitude>
                <eastBoundLongitude>180</eastBoundLongitude>
                <southBoundLatitude>-90</southBoundLatitude>
                <northBoundLatitude>90</northBoundLatitude>
        </EX_GeographicBoundingBox>
        <DataURL>
                <Format>image/gif</Format>
                <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
                                                xlink:href="http://Filter_WebService_Address/" />
        </DataURL>
</Layer>
```

Each data-layer is defined in XML tag element called "*Layer*" in GIS domain. There are some other attributes and sub elements defined by the standard metadata schema in layer element. For the illustration purpose we just give the brief layer element with the most necessary attributes and sub element. Some times we do not even need to use "DataURL" sub elements; instead, we use registry-catalog service in order to find the Filters providing the requested data.

The most important attribute to cascade and chain the filters is the attribute "cascaded". If a WMS cascades the contents of another WMS, then it shell increment by 1 the value of the cascaded attribute for the affected layers. If that attribute missing from the originating server's service metadata, then the cascading WMS shell insert the attribute and set it to 1. If the length of the chain of the filters providing same data increases then, the value of the attribute "cascaded" increases. The value is 0 at the filter providing data from first hand. 0 is the default value for the attribute "cascaded".

WMS at the top of the cascaded filters which IDS tools interact with is WMS 1 in Figure 5. So, general picture or snapshot of the filters and their data provided are displayed by the IDS through WMS 1's capability metadata. It is important Filters keep capability metadata up-to-date. A sample IDS screen is shown in Figure 3 for displaying selected data in layers for the sample filter chaining shown in Figure 5. For the demonstration purposes, layers in red in Figure 5 are assumed to be selected by the user and displayed on the IDS screen. In this document, we do not explain how WMS creates maps or WFS provides data from its Databases or file systems based on queries. We focus on the common architectural issues for distributed data access, query and transformation through capability federation of Web Service based Filter services.

Filters have standard interfaces and therefore, except requests created in accordance with the predefined schema. Filters support “getCapability” routine to enable federation of Filters. Here we give a sample request to get Google map data from the WMS 1. Requests for data are in predefined format.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetMap xmlns="http://www.opengis.net/ows">
        <version>1.1.1</version>
        <service>wms</service>
        <exceptions>application_vnd_ogc_se_xml</exceptions>
        <Map>
                <BoundingBox decimal="." cs="," ts=" ">-124.85,32.26,-113.56,42.75</BoundingBox>
                <Elevation>5.0</Elevation>
                <Time>01-01-1987/12-31-1992/P1Y</Time>
        </Map>
        <Image>
                <Height>400</Height>
                <Width>400</Width>
                <Format>video/mpeg</Format>
                <Transparent>true</Transparent>
                <BGColor>0xFFFFFF</BGColor>
        </Image>

        <ns1:StyledLayerDescriptor version="1.0.20" xmlns:ns1="http://www.opengis.net/sld">
                <ns1:NamedLayer>
                        <ns1:Name>Nasa:Satellite</ns1:Name>
                        <ns1:Description>
                                <ns1:Title>Nasa:Satellite</ns1:Title>
                                <ns1:Abstract>Nasa:Satellite</ns1:Abstract>
                        </ns1:Description>
                </ns1:NamedLayer>
                <ns1:NamedLayer>
                        <ns1:Name>California:States</ns1:Name>
                        <ns1:Description>
                                <ns1:Title>California:States</ns1:Title>
                                <ns1:Abstract>California:States</ns1:Abstract>
                        </ns1:Description>
                </ns1:NamedLayer>
                <ns1:NamedLayer>
                        <ns1:Name>World:Seismic</ns1:Name>
                        <ns1:Description>
                                <ns1:Title>World:Seismic</ns1:Title>
                                <ns1:Abstract>World:Seismic</ns1:Abstract>
                        </ns1:Description>
                </ns1:NamedLayer>
        </ns1:StyledLayerDescriptor>
</GetMap>
```

Why do we need that (capabilities federation)? Why don’t we use first case (accessing the filters separately)? We answer these questions in the following Chapter.

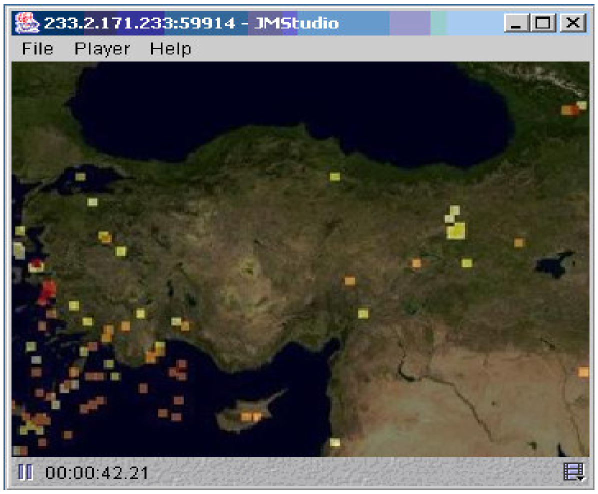


Figure 6: WMS at CGL federates capabilities of WMS at NASA OnEarth and WFS at CGL. WMS and WFS are filters in our context.

3.3. Implications of the Compasibility Nature of the Filters through Capabilities Federation

We create a system composed of Web Services as Filters for the distributed data access, query and transformation. Each filter describes itself by its capabilities metadata. Filters can aggregate and update their capability files dynamically (at run time) or manually (before run time) based on the system configuration. Each filter is both a client and a server for the other filters.

Composing filters through capabilities federation is expected to increase performance. Main reasons for the performance gains are fault tolerant data access, query and transformation, and load balancing. Creation of comprehensible data (can say virtual information-data) through the filters chaining on the way to the final destination is another gain of the filter composing.

Fault Tolerance; if a filter keeps different filter server addresses providing the same data, then in case of one of them fails another one can be connected and data is accessed. There is no single point of failure. Filters keep this information in their capability metadata. For each data there will be multiple online services name and access point information.

```
<Layer cascaded=”1”>
        <name> A </name>
        <DataURL>
                <Format>image/gif</Format>
                <OnlineResource xlink:href="http://Filter_WebService_Address/FILTER-1.wsdl" />
        </DataURL>
        <DataURL>
                <Format>image/gif</Format>
                <OnlineResource xlink:href="http://Filter_WebService_Address/FILTER-2.wsdl" />
        </DataURL>
</Layer>
```

Here is the concrete example from the capability metadata providing redundant filter addresses Filter-1 and Filter-2 for the data A. Filter which owns this metadata part first connects to Filter-1 to get data A and if it fails then, try to access the Filter-2. By using redundant filters we reduce the possibility of failure in data access.

Load balancing; sometimes a data set need to be chunked into small pieces and each piece is obtained from different filters in parallel. Capability metadata can be used to determine how to combine services into filter chains with interconnected input-output ports. This causes lots of performance gains. All these expected performance gains will be proven by the real application tests. Since Filters need to chunk the request into small requests to different successive filters and assemble the returned parts to create whole result, Load balancing is a little bit tricky compared to Fault tolerance at the application level. This needs application specific implementation at each Filter depending on the data attributes. Related metadata is same as in the case of Fault tolerant displayed above.

Comprehensible data creation; normally data kept in the file systems and Databases are raw data, these data can not be comprehended by the end users. So, they need to be processed and transformed to an understandable format for the end users and decision makers to make wisdom decisions. Through filters transforming capability, decision makers can obtain more comprehensible data for their wisdom decisions.

4. Summary and Conclusions

4.1. Summary of Research Issues

Here we summarize briefly what the issues are on our way to create the ASIS framework for the Web Services as federated data and information filters to support distributed access, query, and transformation.

We propose an architecture providing capability federation of Web Services and enabling attribute based querying of information. In that concept, we define the generic and domain specific functionalities of the capabilities in the system to federate the filters. Furthermore, we address the questions of what elements and attributes are needed for metadata and data models (ASL) to facilitate inter-service communications for enabling distributed access, query and transformation of data. In general, metadata is domain specific. However, some metadata crosses many domains, such as information about the creator. Metadata thus need to be able to cover generic as well as domain dependent attributes.

We define the general architecture of the Filter Services. Filter Services are grouped into two categories: ASFS and ASVS. They talk to each other in ASL (such as GML and CML). ASL is a common data representation format. We define requirements and schema for ASL and its functionalities in the system to federate the information sources. We also define the common ports for the Filters to communicate, locate and bind each other to create more complex information and comprehensible representations of any data through inter-service communication.

We demonstrate our system in GIS domain with Web Map Services federation and create scientific data plotting Web Service based Filter Services, and integrate them together to create more comprehensible and informatory data (knowledge or wisdom) for the end users. Scientific data plotting services are DISLIN-based and created according to the proposed Filter Service architecture.

We integrate the ASIS system extendable with the application science simulations (Pattern Informatics, Virtual California and GeoFEST in ServoGrid) through IDS tools. IDS tools are created for utilizing integrated Filter Services for GIS domain. IDS tools enable creation of map animation, map movies, map images, and interactive query support to get further information on the image and/or animation [1, 32].

We plan to enable binding of services into pipelines with or without human intervention through metadata. Currently, we bind them statically before runtime.

In order to handle large scientific data in an efficient and robust manner, we plan to introduce application-level caching and load balancing into the ASIS to handle large scientific data in an efficient and robust manner.

4.2. Expected Contributions

We formalize the Web Services as federated data and information filters to support distributed access, query, and transformation. We call the proposed system ASIS. ASIS enables attribute based querying through capabilities metadata, defining all the data/information sources as interacting Web Service filters with standard metadata service ports. GIS is our motivating domain. We will demonstrate proof of concepts for other domains (such as chemistry and astronomy). We provide capabilities federation through proposed Web Services' capabilities metadata as distinct from data/Database federation/replication approaches. We define the differences of our approaches compared to data federation.

We define requirements and instructions about how to build ASL and metadata in XML based *capability documents* for the application sciences and specific information system (ASIS). We also create an architectural framework for federating multiple filters with capability and speaking ASL.

We define possible bottlenecks and optimization and enhancement opportunities for the distributed heterogeneous information management systems. In that context, the composability nature of our filters enables caching and load balancing for obtaining enhanced service outcomes. Capability aggregation and dynamic capability exchanges and updates are the other issues serving optimization and architecture enhancement purposes.

We also provide enhanced decision support with domain specific metadata languages and interactive mapping tools with query capabilities. We propose an architecture framework to transform heterogeneous and dispersed data into human readable forms (such as maps in GIS) and integrate multiple information sources into interactive user interfaces such as digital photography, demographic information, and information from simulations.

5. Milestones

What we have done:

Implement OGC Web Map Services (WMS) [7]

Implement OGC Web Feature Services (WFS) (*implemented by Galip Aydin*)

Create IDS Tools: It is interactive and modular. It is extendable for any other geophysics applications.

OGC compatible GIS system for accessing, querying and transforming spatially referenced geographic data/information through Web Service-based Filter Services.

Create ASIS architecture for GIS domain: It is composed of Web Services based Filter Services in the form of ASFS (extended from WFS) and ASVS (extended from WMS). ASIS is inspired from OGC.

Integrating ASIS with IDS tools and the other Lab projects (HPSeach and FTHPIS) for the geophysics applications from ServoGrid.

Testing and experimenting whole system with real geophysics applications from ServoGrid. We create maps, movies and animations from the set of data coming from the distributed heterogeneous data providers.

What we will do:

Formalize the two fundamental components of the ASIS (capabilities and ASL) and define Web Services as federated data and information filters from our experience of GIS domain. The capabilities should enable distributed data access, query and transformation through inter-service communication.

Enhance ASIS by adding caching load balancing capabilities to the Filer Services.

Implement Scientific Plotting Services according to proposed Filter Service architecture and integrate with ASIS: We will develop a DISLIN-based [33] Web Service-based Filter Service that takes tabular data and returns a comprehensible data formats in images such as JPEG.

Define when our approach is ok when it is not: Based on our test of GIS-based services, we will analyze the other domains for their suitability to our approach. We will formalize the requirements for creating capability federation of Web Services.

Compare the system (ASIS) with other related systems (based on distributed data federation)

Tests and experiments: We will demonstrate the basic functionalities of our system and our capability-based approach by doing interactive data access and querying through IDS tools and ASIS architecture in GIS domain.

REFERENCES

[1] Project home page http://complexity.ucs.indiana.edu/~asayar/gisgrids/

[2] Open Geospatial Consortium (OGC) web site http://www.opengeospatial.org/

[3] Roy Williams et al., “Virtual Observatory Architecture Overview”. Version 1.0 Document # IVOArch-20040615, IVOA Note 15 June 2004.

[4] Liping Di, “Distributed Geospatial Information. Services-architectures, Standards, and Research Issues”, XXth ISPRS Congress, 12-23 July 2004 Istanbul.

[5] Simon Cox , Paul Daisey, Ron Lake, Clemens Portele, Arliss Whiteside, Geography Language (GML) specification 3.0, Document #02-023r4., January 2003.

[6] Vretanos, P. (ed.), Web Feature Service Impl. Specification (WFS) 1.0.0, OGC Document #02-058, Sept 2003.

[7] Jeff De La Beaujardiere, OpenGIS Consortium Web Mapping Server Implementation Specification 1.3, OGC Document #04-024, August 2002

[8] Mehmet Aktas, Galip Aydin, Andrea Donnellan, Geoffrey Fox, Robert Granat, Lisa Grant, Greg Lyzenga, Dennis McLeod, Shrideep Pallickara, Jay Parker, Marlon Pierce, John Rundle, Ahmet Sayar, and Terry Tullis “iSERVO: Implementing the International Solid Earth Research Virtual Observatory by Integrating Computational Grid and Geographical Information Web Services” Technical Report December 2004, to be published in Special Issue for Beijing ACES Meeting July 2004.

[9] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver and K. Glover, M.R. Pocock, A. Wipat, and P. Li. Taverna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics, 20(17):3045-3054, Oxford University Press, London, UK, 2004.

[10] Ewa Deelman, Gurmeet Singh, Malcolm P. Atkinson, Ann Chervenak, Neil P Chue Hong, Carl Kesselman, Sonal Patil, Laura Pearlman, Mei-Hui Su, ”Grid-Based Metadata Services”, SSDBM04, Santorini, Greece, June 2004.

[11] Singh, G., S. Bharathi, A. Chervenak, E. Deelman, C. Kesselman, M. Manohar, S. Patil, and L. Pearlman. A Metadata Catalog Service for Data Intensive Applications. Proceedings of the Supercomputing Conference, November 2003.

[12] M. Radenkovic and B. Wietrzyk, "Life Science Grid Middleware in A More Dynamic Environment", In Proc. of Grid Computing and its Application to Data Analysis (GADA'05), Ayia Napa, Cyprus, 2005, LNCS, pp 264-273.

[13] Carole Goble Putting Semantics into e-Science and Grids in Proc E-Science 2005, 1st IEEE Intl Conf on e-Science and Grid Technologies, Melbourne, Australia, 5-8 December 2005.

[14] National Virtual Observatory (NVO) quarterly-report “Building the framework for the National Virtual Network”, April-June 2006. Project web site is available at http://www.us-vo.org/

[15] Fran Berman, Geoffrey C, Fox, Anthony J. G. Hey., Grid Computing: Making the Global Infrastructure a Reality. John Wiley, 2003.

[16] Foster, I. and Kesselman, C., (eds.) The Grid 2: Blueprint for a Computing Infrastructure, Morgan Kaufmann2004.

[17] VOTable: A Proposed XML Format for Astronomical Tables, version 1.0, R. Williams, F. Ochsenbein, C. Davenhall, D. Durand, P. Fernique, D. Giaretta, R. Hanisch, T. McGlynn, A. Szalay and A. Wicenec, 15 April 2002.

[18] F. Curbera, Y. Goland, J. Klein, F. Leymann, D. Roller, and S. Weerawarana Business Process Execution Language for Web Services, Version 1.1. Specification, BEA Systems, IBM, Microsoft, SAP, Siebel, 05 May 2003. http://msdn.microsoft.com/library/default.asp?url=/library/enus /dnbiz2k2/html/bpel1-1.asp

[19] Interactive Decision Support (IDS) Tools User Manual for ServoGrid Applications is available at http://complexity.ucs.indiana.edu/~asayar/gisgrids/docs/usermanual.pdf

[20] Beth Plale , Jay Alameda , Bob Wilhelmson , Dennis Gannon , Shawn Hampton , Al Rossi , Kelvin Droegemeier, Active Management of Scientific Data, IEEE Internet Computing, v.9 n.1, p.27-34, January 2005.

[21] Zaslavsky, Ilya, Ashraf Memon, “GEON: Assembling Maps on Demand from Heterogeneous Grid Sources”, Twenty-fourth Annual ESRI International User Conference, August 9-13, 2004, San Diego, CA.

[22] Bhatia, Karan, Ashraf Memon, Ilya Zaslavsky, Dogan Seber, Chaitan Baru, “Creating Grid Services to Enable Data Interaoperability: Example from GEON Project”, GSA Annual Meeting in Seattle, WA November 2-5, 200.

[23] OGSA-DAI Open Grid Services Architecture Data-Access and Integration http://ogsadai.org.uk

[24] Mario Antonioletti, Neil Chue Hong, Ally Hume, Mike Jackson, Amy Krause, Jeremy Nowell, charaka Palansuriya, and Martin Westhead. Experiences of designing and implementing grid database services in the ogsa-dai project. In Global Grid Forum Workshop on Designing and Building Grid Services, September 2003.

[25] Rajasekar, A., M. Wan, and R. Moore, (2002), “MySRB & SRB – Components of a Data Grid,” The 11th International Symposium on High Performance Distributed Computing (HPDC-11) Edinburgh, Scotland, July 24-26, 2002.

[26] Implementing and using SRB, Proc UK e-Science All Hands Meeting 2003, © EPSRC Sept 2003, ISBN 1-904425-11-9.

[27] MCAT, (2000) "MCAT:Metadata Catalog", SDSC. Available at http://www.sdsc.edu/srb/index.php/MCAT

[28] Harshawardhan Gadgil, Geoffrey Fox, Shrideep Pallickara, Marlon Pierce, Robert Granat A Scripting based Architecture for Management of Streams and Services in Real-time Grid Applications Proceedings of the IEEE/ACM Cluster Computing and Grid 2005 Conference (CCGrid 2005). Cardiff, UK May 2005 Pages 710-717.

[29] Mehmet S. Aktas, Geoffrey C. Fox, Marlon Pierce Fault Tolerant High Performance Information Services for Dynamic Collections of Grid and Web Services FGCS Special issue from 1st International Conference on SKG2005 Semantics, Knowledge and Grid Beijing China November 27-29 2005

[30] Ackoff, R. L., "From Data to Wisdom", Journal of Applies Systems Analysis, Volume 16, 1989 p 3-9

[31] Michael Wan, Arcot Rajasekar, Reagan Moore, Phil Andrews, "A Simple Mass Storage System for the SRB Data Grid", Proceedings of the 20th IEEE Symposium on Mass Storage Systems and Eleventh Goddard Conference on Mass Storage Systems and Technologies, San Diego, April 2003.

[32] Sayar A., Pierce M., Fox G., OGC Compatible Geographical Information Services Technical Report (Mar 2005), Indiana Computer Science Report TR610.

[33] Scientific Data Plotting Services (DISLIN), project page is available at http://www.mps.mpg.de/dislin/

[34] Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., and Orchard, D. “Web Service Architecture.” W3C Working Group Note, 11 February 2004.

[35] Sayar A., Pierce M., Fox G., Developing GIS Visualization Web Services for Geophysical Applications, ISPRS 2005 Spatial Data Mining Workshop, Ankara, Turkey.

[36] Sayar A., Pierce M., Fox G., Integrating AJAX Approach into GIS Visualization Web Services. IEEE International Conference on Internet and Web Applications and Services, ICIW'06, February 2006.

[37] E. Deelman, et al. "Grid-Based Metadata Services," 16th International Conference on Statistical and Scientific Database Management (SSDBM), Santorini, Greece, 2004.