Ahmet Sayar

Summary of Thesis Proposal

1. Statement of the Problem and Relevancy

In Science domains information/data is inevitably distributed among several data resources. Science applications need to access and integrate data and specialized computational capabilities (visualization, statistical analysis, data mining etc.) of wide range of relational and un-relational data sources.

Accessing and integrating data from different providers using different heterogeneous technologies is a difficult task which has been worked upon for decades. 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.

The literature shows many proposals for integration of data, ranging from federated databases to mediators (such as SRB-Storage Resource Broker) and ontology. The adoption of a conventional integration methodology does not lead to a solution for comprehensible data integration. They do not consider the graphic aspects to represent schemas nor the diversity and richness of semantic representation of the data. One can achieve data integration to some extent by using conventional techniques but one can not integrate the data at the representation level. We call the representation level as layer-level (or view-level) and propose an architectural framework to integrate data at that level.

In addition to this main problem, integrating distributed and heterogeneous Science data at the presentation level and mapping them with Science Grid applications require quick response times. The presentation level integration framework must enable accessing processing of large data sets in a reasonable time period. This is a common problem of data integration proposals with respect to performance. In the present report we focus on performance issues at the layer-level.

In summary, our main goal is creating a general framework mapping distributed and heterogeneous data resources into global Science Grids in a performance efficient manner. To achieve this, we need to access and integrate the data and present it to the Science Grids in accordance with their general needs. We also need to create a general view-based integration interface for clients to utilize data, enable integration and interoperation of data and Science Grids at view-level (layer-level) by hiding the complexities of the system.

2. Context of Proposed Research in Literature

Other systems such as SRB (Storage Resource Broker) and FDBS (Federated Databases Systems) have examined various general approaches to data federation. Unlike them we propose an alternative solution based on what we call “capability” federation of Web services at the view-level and, leverage third party data-integration approaches at the bottom level (see Figure 1).

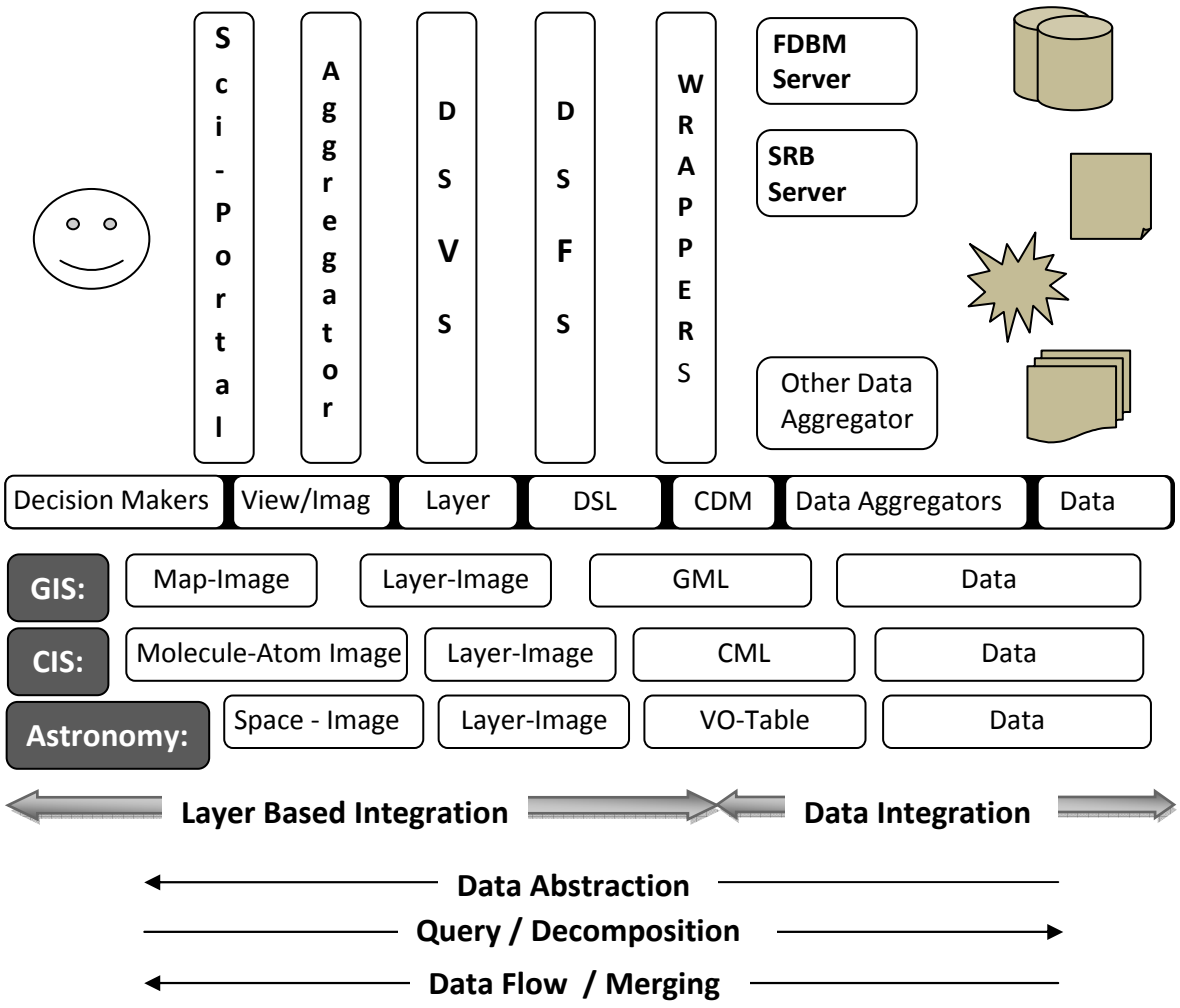
At the view-level, data is abstracted as layers. Layers are also abstracted to final representation before accessing to the decision makers. Its name depends on the domain. In our implementation domain we call the final representation as map. It might be called observatory or space in Astronomy and, molecules or atoms in Chemistry.

In our approach, we introduce two new types of services. These are DSVS and DSFS which are inherited from OGC’s standards, but we enhanced them by implementing in Web Services principles. Web Service enables them to be loosely coupled, chained and easily located by registering to catalog services. Each service is described by generic and domain specific metadata descriptions (capability) and a Web Service Description file (WSDL) that can be queried through Web Service invocations. DSVS and DSFS are capable of exchanging and federating their capabilities dynamically. This helps to chain services and workload management. 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.

The proposed system presents data to users by means of user views that, from a user perspective, looks exactly like the views of data in a DSVS aggregator (we call it aggregator WMS). User views in a DSVS aggregator are mapped to underlying data objects that may be stored on any of the various DSVS or/and DSFS in communication. Communication links are built based on capabilities-metadata. DSVS and DSFS have their own capability-metadata in a standard schema. DSVS aggregator processes queries in terms of these views, retrieving data as needed from the other systems in the federation, and delivering the results as if the entire data is local.

The DSVS aggregator is actually a DSVS (for the sake of interoperability) with cost-based optimizer that is aware of the distribution and heterogeneity of the back-end servers. In order to deliver an acceptable performance we also propose an innovative load balancing with caching techniques (see the section 3.xx and Figure xx for the GIS case). DSVS aggregator decomposes each query into sub-queries that can be handled by the various DSVS and DSFS in the federation, sends each sub-query to the appropriate DSVS or DSFS, and merges and delivers the overlaid result layers to the user through Science portal. Science portal interface and available data layers are created according to Aggregator DSVS’s capabilities-metadata.

We use wrappers (or adaptors) for leveraging other data integration systems (such as SRB and FDBS) and any other third party data resources. A wrapper defines the communication mechanism that will be used to access data and from a remote data source. More specifically, it is a mechanism that the DSFS and DSVS servers (Aggregated WMS in our framework or FDBM in FDBS) use to communicate with federated servers of other data integration systems. Since we use OGC compatible servers for our GIS framework we need the wrappers just for interacting with non-OGC servers. We also wrap Google Map and Dislin Plotting libraries as DSVS and integrate them to the Science Grids. In case of systems using wrappers; if the source changes, the adaptor may have to change, but application may never see it. Furthermore, adding a new source is easy, a new adaptor may need to be written, or even it may already be exist online.



CDM: Common Data Model - Global application schema, DSL: Domain Specific Language -Structured Common Data Model

Figure 1: Proposed DSIS Architecture integration at view and data levels. This is basically details of orange rectangle in Section 3.x Figure 2.

2.1. Locating the work in Literature

In summary our research focuses on creating Grid-oriented view-services integrating data at the view-level and providing them to Science Grids through web portals with interactive-smart view tools. Web portals enable Grids simulation can be accessed in a remotely and run seamlessly. Further more interactive-smart map tools hide the complexities of the system and make it easy to be used by scientist or end-users from different backgrounds. We also associate the inputs and outputs of the Science Grid simulations at the view level of data integration hierarchy`and, create three-layer structured views illustrating the association of input and outputs of the Science simulations for the decision makers and end-users.

The data access and integration part of our system shows similarities to the FDBS and the SRB. However, it has some differences in the level of data/information integration hierarchy.

Federated Database Systems (FDBS): A federated database enables communication between multiple DBMS or databases in a single SQL statement and makes the location of information transparent to the user. Constituent databases are geographically decentralized and integrated remotely. A federated database (or virtual database) is fully-integrated. To achieve this a logical composite of all constituent databases in a federated database system used.

Federated Database (or virtual database) in DBMS corresponds to Aggregator DSVS in our integration framework. They both serve for the same purposes.

DSVS and our system both make data abstraction. Data abstraction in our framework is representation based and leveraged. For example, for GIS domain; any data abstracted as GML, and GML is abstracted as layer, and layer is abstracted as map images, even map images are abstracted by overlaying on each other.

Regarding data integration, this depends on the data abstraction techniques used. In order to integrate data, FDBS uses mapping of the data base views represented in schema. The mapping techniques can be generalized into two groups. In “Global as View (GAV)” the global schema is defined in terms of the underlying schemas. In “Local as View(LAV)”, the local schemas are defined in terms of the global schema. Since we use global schema for the data abstractions (GML for feature data and image types for layers and maps) our approach looks like LAV.

Regarding the components integrated, FDBMS integrates data by federating databases. We not only integrate the databases but also any other data servers such as WFS, WMS or even SRB servers. To this end, we are relatively located at the upper level of the data integration hierarchy. Our next step will be applying this approach to any other domain by adding domain specific layer to top of the current data integration architectures such as SRB and FDBS.

Regarding data-flow in the system, FDBS provide two-way data flowing (transaction management). In contrast, we enable just one-way data flow (data querying). In our approach, data querying is done by decomposition based on the information kept in capabilities metadata. That is, query is decomposed to

sub-queries and each sub-query is sent to corresponding data server and, result sets are matched as an answer to the main query.

SRB (Storage Resource Broker): SRB is the implementation architecture of the integrated/federated digital libraries. They integrate data as digital objects. Digital objects are mostly files but URLs and SQL command string and any string of bits collected from multiple different data sources. Digital objects are geographically decentralized and integrated in a remote fashion.

SRB doesn't define metadata in XML structure they store metadata in MCAT as relational databases. We use XML structured capability metadata in distributed fashion. That is, each server keeps its metadata locally and upgrades its capability metadata dynamically through inter-service communication capability of our system. Inter-service communication is achieved by separate ports and standard services (getCapabilities) enabling exchanging and federating other servers' capabilities metadata.

Regarding data-flow in the system, SRB provide two-way data flowing (transaction management). In contrast, we emphasize one-way data flow (data querying). SRB has central metadata handling approach. Instead, we have distributed metadata handling approach.

We use two level data integration (view-level and data-level). We leverage SRB server into our data-level integration by using wrapper. SRB is not competing but a complementary technology. As it is shown in the figure we use wrappers to integrate SRB servers to the system.

We just deal with unified data access and query capabilities but SRB is dealing with data transferring and replicating as well as data access and querying.

Capability concept in literature: We use capabilities defining service and data in view-level. It enables inter-service communication through well-defined service interfaces and message formats ("getCapabilities"). It is updated manually or dynamically. It consists of descriptor, service and provider metadata. Inter-service communication is achieved without a third-party and enables chain of services. In our system we propose two different groups of servers, DSVS and DSFS. These have different capabilities schema and formats. Communications between servers are achieved through exchanging the capability documents.

Capability is inspired from OGC WMS capability specification. It looks like Dublin Core metadata format defining resources. Capability like structure is also used in Gannon's approach (XPOLA), for Grid services' security issues, describing dynamic Web/Grid resources.

3. Problem Solution in GIS domain

Grid-oriented map services and their integration to Geo-Science Grids

<< Previous document goes in HERE – with some updates at Section 2 as Geoffrey commented >>

http://complexity.ucs.indiana.edu/~asayar/proposal/CurrentResearchWork1.pdf

4. Expected contributions

We have developed a generic information management framework called Domain Specific Information Systems (DSIS), Grid-oriented DSVS (specifically WMS in GIS domain) and integration framework for the DSVS and Science Grids. DSIS and integration framework can be applied to any domain with their domain specific and predefined DS-(Languages, capabilities and, VS and FS services). We also define instructions and requirements how to build DSIS by defining DSVS, DSFS and DSL. We also formalize DSIS supporting distributed access, query, and transformation through capabilities metadata, defining all the data/information sources as interacting Web Services with standard metadata service ports.

GIS is our motivating domain. 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 and leverage their systems into DSIS. We make distinction of data integrations based on the data-lifecycle in the integration hierarchy.

We define possible bottlenecks and optimization and enhancement opportunities for the distributed heterogeneous information management systems. In that context, the compos-ability nature of our DSVS and DSFS enable caching and load balancing for obtaining enhanced service outcomes. Capability aggregation, 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.

More Specifically for GIS Domain: We merge two important software worlds: GIS and Web Service Architectures, we extend OGC capability specifications with the workload management, we create an architectural framework for dynamic capability federation of Map Services enabling workload management and service chaining, we create Web Service-based scientific plotting services and Map Services and, integrate them into Science Grids, we create "smart map" tools as portlets for the Science portals, we design high performance Web Service architecture for distributed Map Services to support archived geospatial data, we create scientific plotting tools in Web Service principles for Geo-science Grids, we create Grid-oriented Map Services and create an architectural framework to integrate Map Services with the Geo-science Grids.

5. Future Work

We mostly focused on structured (different data models), syntactical (different languages and data representations) and systemic (different hardware and operating systems) heterogeneity in data integration issues at the view-level. Since we applied our proposed integration framework in OGC Compatible GIS domain we did not take the semantic heterogeneity of the data into consideration. General data model and semantics of the data are defined by OGC. We will be extending our framework by researching on semantic issues to make our framework applicable over all Science domains such as Astronomy and Chemistry.

For our proposed DSIS our motivating domain was GIS and, we have used GML for DSL, WMS for DSVS, WFS for DSFS, and OGC's *capability* definitions for capability metadata of DSVS and DSFS services. When we try to apply the framework to other domains such as Astronomy domain, we will need to be using SkyNode as DSFS, VOPlot and TopCat as DSVSs, VOTable and FITS as DSL, and VOResource as capability metadata. Regarding Chemistry domain, there is no chance to find standardized servers corresponding to DSVS, DSFS and capability metadata but they have a standard DSL called CML. We will be using CML as DSL and may be JChemPaint as DSFS but we need to introduce a new online service definition for DSFS. We also need to demonstrate proof of concepts for these domains.

As it is explained above, whenever we need to apply the proposed framework to any other domain, we have to find the corresponding services and general data model in order for our framework to work. Instead of doing this repetitive work for different science domains, we want to create general definitions of DSVS and DSFS applicable to any domain. So, from our experience so far with GIS domain, we see that we need to use ontology and semantic approaches to solve the data integration issues and creating more generic framework working over various Science domains.

We also need to enable binding of services into pipelines with or without human intervention through metadata by using Catalog services. Currently, we bind them before run-time by manipulating capabilities metadata of DSVS and DSFS.