P1.31 A SOFTWARE SYSTEM TO SUPPORT EARTH SCIENCE DATA SERVICES

Yuechen Chi *, Ruixin Yang, and Menas Kafatos
George Mason University, Fairfax, Virginia

Abstract

In Earth science applications, data services, such as data mining, subsetting, reformatting, and resampling, become more and more important primarily due to the ever-increasing large volumes of data with heterogeneous formats, spatial-temporal coverage and resolutions, and at distributed locations. Existing data systems are lacking support for these data service related activities. We have been developing a data and process management and data processing support system for various data service oriented applications. We have also developed standards for the development of data services, and communication protocols between the data services and the system. This system enables data service providers to easily develop data services and register the data services they provide into the system, as well as enables their users to easily utilize the services to process data represented in the system. In this paper, we describe the system architecture, software components, data model, functionality, and user interfaces of the software system. We also use a fully functional prototype to demonstrate the capabilities of the system with example data services for rainfall data from Tropical Rainfall Measuring Mission (TRMM).

1. Introduction

In research and applications using Earth science data, users need to search to find data, retrieve to get the data, and analyze the data to discover and apply knowledge. With advanced satellite global observations and high-performance model simulations, they face the challenge of large volumes of datasets, large size of data objects, heterogeneity in data formats and spatial-temporal coverage and resolutions, and wide data distribution. For example, NASA Earth Observing System (EOS) is putting together a series of satellites from 1997 to 2008, and produces large volume of data products at a data rate of 1.3 TB/day, while it is 20 GB/day only for TRMM. The data products are stored in 7 Distributed Active Archive Center (DAACs) around the nation.

* *Corresponding author address*: Yuechen Chi, George Mason University, Center for Earth Observing and Space Research, School of Computational Sciences, Fairfax, VA 22030-4444; e-mail: ychi@mason.gmu.edu

Many data systems support metadata-based data search by which users can search data by dataset, time, location, etc. For example, a typical example might be “find the TRMM data from 1999-09-06 to 1999-09-20 and covering the region of North Atlantic Ocean". However, many users would like to search out archived data with interested phenomena or based on value-range of physical parameters, for example, "find the TRMM data containing Hurricanes" and "find the TRMM data with rainfall rate greater than 4 mm/hour". Many data systems support direct download of original data with some pre-defined formats. But very often users are interested in only a fraction of the data they received, the data formats are not accepted by their analysis tools, and the data resolutions do not match their specification. Many data systems provide users browsable images of their interested data. But users sometimes need to perform online analysis on massive data from an archive for certain applications.

Data services, such as data mining, subsetting, reformatting, and resampling, are further needed to apply on the data along with data search, retrieval and analysis for users to find data precisely they need, prepare the data for use, and discover knowledge. Existing data systems lack support of these data services.

Many data systems handle the data requests by pulling out data from an archive and copying the data to computer media for delivery or staging the data in a computer for users to ftp. In these data systems, users have to download the data to their local computing environment and perform the data services outside of the data providing system. This results in massive data migration, and time and resource consuming in the development of custom code for the management of data and data processing.

With dramatic technology advancing of online storage and Internet networking, more and more data are immediately and easily available and accessible. For example, NASA's Earth Observing System Data and Information System (EOSDIS) project is implementing data pools, which are large, online, and smart disk farms. Data pools contain large amount of observational data online for users to directly and rapidly access. This enables the development of online data services in data systems, in which the data systems process data represented in the system on-the-fly according to user requirements and deliver results to them directly.

Many data systems are driven by database management systems (DBMSs). Many DBMSs support user-defined functions, in which users can register and invoke user-defined functions in the DBMSs. Data services can be developed into user-defined functions in DBMSs to be applied on the data in the system. DBMSs usually have proprietary database programming language for the development of user-defined function, such as a stored procedure. However, Earth science data services are often too complicated to be developed with present database programming language. Many DBMSs also support the development of user-defined functions with certain programming languages, such as C. But many Earth science applications are written with programming languages widely used in Earth science community but not supported by DBMSs, such as Fortran and IDL, and it is difficult and resource consuming to convert the applications into the supported programming languages.

Data service providers usually integrate the data services in the middleware layer of data systems. However in current data systems, the data services are tightly coupled with the data system, and data service providers need to have access and thorough knowledge of a system to integrate a data service into the data system.

At George Mason University, we have been developing a data and process management and data processing support system for the data service oriented applications [1,5,6,7]. In this system, the data services are loosely coupled with the system. We also developed standards for the development of data services, and communication protocols between the data services and the system. In this system, data service providers can develop a data service with any kind of general purpose programming language or convert their applications written by any programming language into a data service following the standard. This system also enables data service providers to easily supply a data service into the system and their users to easily utilize the service to act upon the data represented in the data system in a plug-and-play fashion.

In Section 2, we introduce the system architecture, software components, data model, functionality, and user interfaces of the system we have developed. In Section 3, we describe the standards for the development of data services, and communication protocols between data services and the system. In Section 4, we describe a web-based prototype system with an example of data services for rainfall data from TRMM. Section 5 describes the implementation status of the system. We give conclusions and further developments in Section 6.

2. System

The overall objective of the system development is to build a computing environment for data service providers to easily supply data services and for their users to easily utilize the data services. The designing and development of system adopt the concept of object-relational technology [4,12,13]. The system uses relational data model for the management of information of data, process, and data processing. It enables users to register a specific data service and invoke it to process the data represented in the system. Within the system, the data applications and interfaces are remotely accessible and can be easily customized. The users do not need to know where the distributed data objects and data service processes reside.

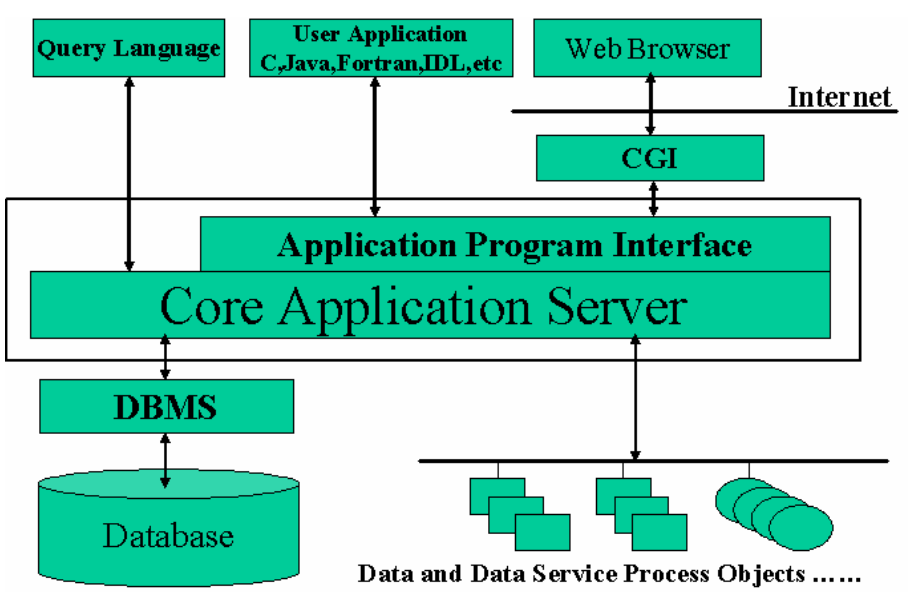

Figure 1. System Architecture

The functionality of the system is designed following the workflow of scientific applications in using Earth science data, namely data search, data retrieval, and data analysis. The database of the system contains a series of tables to store the information of datasets, data objects (granules), data service process and their parameters, and data and process relationships in supporting the functions of the system.

2.1 System Architecture

The system was built on standard 3-tier client/server architecture. Figure 1 shows the system architecture.

Tier 1 represents the individual user clients. The system provides user interfaces of web, API, and query language for users to communicate with the system in this tier. In the web-based interface, users provide and retrieve information, and utilize the data services through a standard web browser. In the query interface, users interact with the system through an interactive program that accepts a user query, sends the query to the system to execute, obtains results, and delivers the results to users. The system also provides users an Application Program Interface (API) for them to seamlessly embed the functions and features of the system into their own applications written by Fortran, C, Java, and other third generation programming languages (3GL).

Tier 2 includes an application server, the core component of the system. It contains the application logic of the system. The application server has three major components of database access, data and process object access, and data service. The database access component stores and retrieves the information of data, process, and data processing in the underlying database through the database management system. The data and processes are objects visible by the system. They can be local files, or remote objects accessible via HTTP or FTP protocol. The data and process object access component allocates and retrieves the data and process objects for data services. The data service component invokes the process to act upon the data represented in the system according to user request, and handles the communication between the data services and the system. The application server accepts user requests from tier 1, processes the requests mainly through the three components, and delivers the results to users in tier 1. Tier 2 also includes an Application Program Interface module that handles the communication between the API call of user applications and the response of the application server.

Tier 3 provides the persistent data, process, and information repository. It includes database storing information of data, process, and data processing, and the database management system. It also includes data objects and process objects, and the file systems and protocols for the access of the objects.

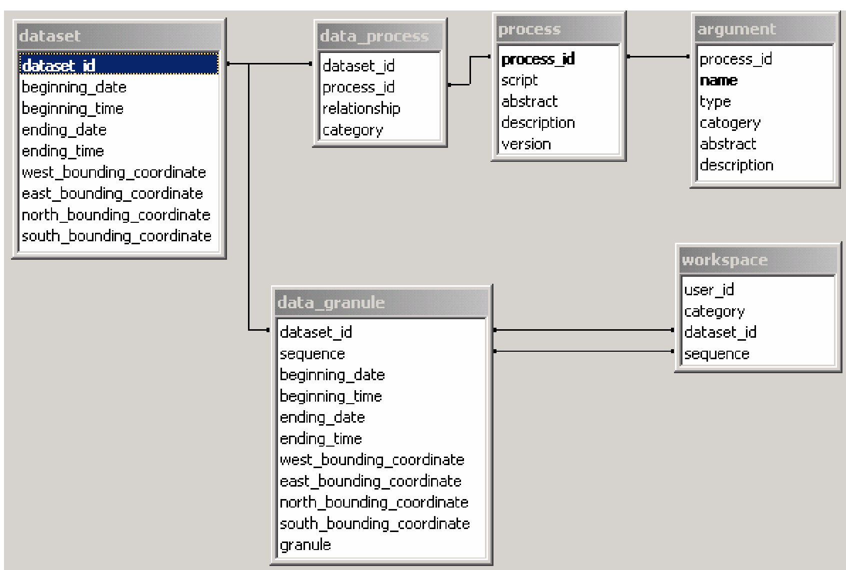

Figure 2. Database Schema

2.2 Data Model

The system uses a relational data model for the management of information about data, data service process, and data processing. The system stores the information in a database. Figure 2 shows the database schema. It contains the following major tables in the database.

The dataset and data_granule tables are used to store the information about data. The dataset table records the dataset identification, temporal period of date and time, and geospatial coverage of latitude and longitude of datasets. The data_granule table records the dataset identification, the sequence of data granule in a dataset, beginning and ending date and time, latitude and longitude boundary, and the location for all of the data granules.

The process and argument tables are used to store the information about data service process. The process table records the process identification, full path of process script, abstract, description, and version of processes. The argument table records the name, type, abstract, and description of all of the arguments for each process.

The data_process table stores the relationship between a process and a dataset. It provides information about what data service processes are used for a dataset, and what type of activity they support, data search, data retrieval, and/or data analysis for the dataset. It also describes data objects of a dataset to be input, output, or return result of a process.

The database also contains a workspace table. It records the information of data objects that users currently work on. It is further categorized into search, retrieval, and analysis workspace to track the data objects that users work on in data search, retrieval, and analysis stage. The system supports multi-user applications. In workspace table, there is a field of user identification to distinguish the data objects for each individual user.

2.3 Functionality

The system is a data and process management and data processing support system in support of data service oriented applications. The top level functionality of the system is designed following the workflow of the data service activities with database loading, database browsing, data search, data retrieval, and data analysis. The database loading function enables users to register information of data, process, and data processing into the database of the system. The users use database browsing function to retrieve the information stored in the database.

The basic task of the system is to allocate the right data service process to act upon the right data objects represented in the system according to user requests along data search, retrieval, and analysis service in their applications. Figure 3 shows the dataflow and data service workflow. The system enables users to utilize the data search process to examine the content of data granules to find the data precisely they need, and stores them in search workspace. The users can download the data objects in the search workspace, or send the data to retrieval workspace. The system enables users to invoke the data retrieval process to get the data in their desired form. The users can also send the data objects in the retrieval workspace into data analysis workspace. The system further enables users to perform online analysis using the data analysis process on the data objects in the analysis workspace, and delivers the analysis results to the users.

The system stores all the information along the activities in the database, and enables users to query about them at any point in the workflow. For example, users can query the data objects in search, retrieval, and analysis workspace. Users can query the data service processes associated with the data objects. The system also supports the retrieval of the data and process objects at any point in the workflow for performing the data service.

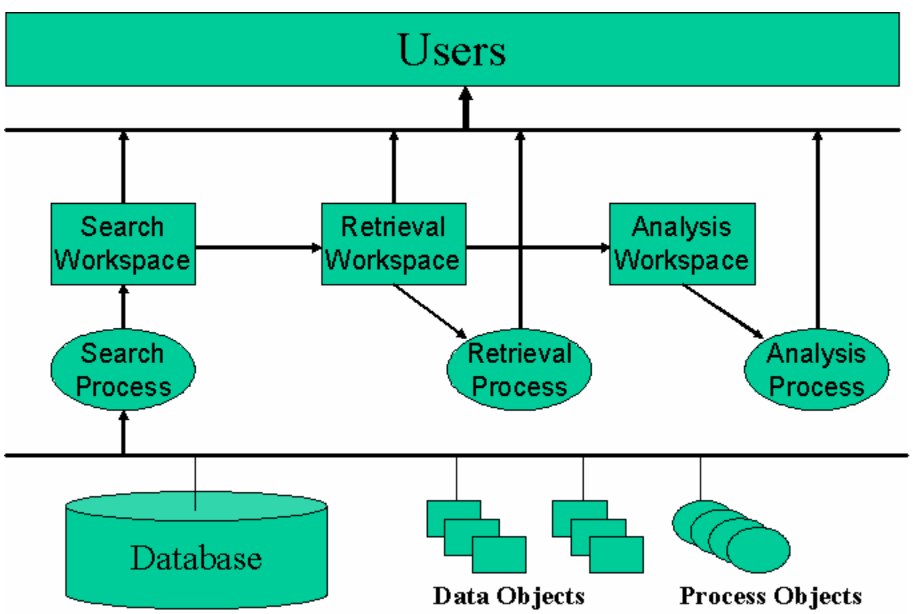

Figure 3. Data Service Workflow

2.4 User Interface

The system provides user interfaces of Web, API, and query language to accommodate the needs of different sectors of the user community. In the web-based user interface, users interact with the system using a web browser from their computers. The system provides users HTML form for collecting the information of data, process, and data processing over the Internet, and delivers HTML pages to users for displaying the results. When users send information via HTTP, the web server receives and passes the information to CGI application. The CGI application passes the information to the application server to store in the underlying database. When users make a request, the CGI application sends it to the application server to obtain the answer, and delivers the results back to the users. The system also enables users to invoke the data services through the web browser. The CGI application then requests the application server to invoke the data service to process the data, generates the result pages dynamically on-the-fly, and finally sends the result back to the web browsers of users.

In the query language user interface, users interact with the system through natural language oriented queries. The system has an interactive program that accepts user queries via the standard input, and then parses the query into specific instructions for the system to execute. After execution, the system sends the result back to the interactive program, and the program delivers them to the users through the standard output. The users have more flexibility to communicate with the system in the query language interface than in the web-based interface. In the web-based interface, users can only construct limited queries with the specific forms. In the query interface, the users can specify any query accepted by the system. We adopted and developed an object-relational query language (ORQL) with SQL dialect for the system.

The system also provides the API interface that enables users to seamlessly embed the functions and features of the system into their applications developed with third generation programming languages, such as Fortran, C, Java, etc. For example, Earth science modeling activities are becoming more data driven. Earth science models are usually developed with Fortran and C programming languages, and run on high-performance computer systems in non-interactive mode. The modeling scientists need a system for them to efficiently locate, retrieve, and analyze the data. They also require that the functions and features of the system can be called from their models and utilize the system non-interactively. Therefore, their models can keep running in a high-performance computing environment.

3. Data Service Development Standard and Communication Protocol

Data service is a process that can act upon the data represented in the system. In our system, the data service is a program with some special types of input and a few strict rules on program output. It is similar to Common Gateway Interface (CGI) programs in the web architecture. Users can develop a data service by any kind of general purpose programming language, such as C, Fortran, Java, etc. as long as following the standard of data service development. When users require a data service from the system, the system will invoke and interact with the data service with the communication protocol between the data service and the system.

The data service is a process with name, parameters, and return result. It gets its parameters through a series of environment variables in the operating system, arg1, arg2, arg3, etc., depending upon their positions in the sequence of arguments. The process returns results through the standard output of the operating system. For convenience, we use arg0 to denote the return result of a process. Users can register a data service into the system with the information of the process and their parameters. The system stores the information in process and argument tables.

For example, we developed "trmm_retrieve" program for data retrieval of TRMM rainfall data. The program reads in a TRMM data granule, extracts a subset within a specified region, converts the result into ASCII format, and returns the URL of the resulting data file. The program is developed with the C programming language with 5 parameters of data, minlat, maxlat, minlon, and maxlon, representing data granule file and latitude and longitude bounding coordinate. The program gets the 5 parameters from environment variables, arg1, arg2, arg3, arg4, and arg5, and prints the result through the standard output. We compile the program, and register the compiled executable process as a data service into the system as:

```
DataURL
trmm_retrieve(data, minlat, maxlat, minlon, maxlon)
```

In our system, the data services are invoked by spawning another process and loading the data service process into the newly created process. After that, the system interacts with the newly created process for providing input parameters and collecting results. The communication between the data service and the system is through the environment variables and standard output. When users invoke a data service, the system first sets a sequence of pre-defined environment variables, arg1, arg2, arg3, etc. according to user supplied parameters. The system then invokes the data service process and passes these environment variables to the data service as its actual parameters. The data service then starts to process the data represented in the system, and sends the results back to the system through the standard output. The system finally delivers the results back to the users.

For example, the “trmm_retrieve” data service can be invoked from a query statement like:

```
SELECT date, trmm_retrieve(granule,20,25,-70,-65)
FROM data_granule
WHERE dataset_id = 'trmm_3b42'
AND date >= '1999-09-06'
AND date <= '1999-09-08'
```

The system will first retrieve the data granules from the data_granule table in the databas, and then invoke the "trmm_retrieve" process 3 times with the actual filename of the data granules.

```
trmm_retrieve('3B42.990906.5.HDF',20,25,-70,-65)
trmm_retrieve('3B42.990907.5.HDF',20,25,-70,-65)
trmm_retrieve('3B42.990908.5.HDF',20,25,-70,-65)
```

For each invocation, the system sets 5 environment variables with the actual values of the parameters and passes the environment variables to the "trmm_retrieve" process. The process gets the actual values of data granule, minimum and maximum latitude and longitude from the environment variables, arg1, arg2, arg3, arg4, and arg5. It then reads in the data granule, extracts the subset of data within the region of latitude 20°N to 25°N and longitude 70°W to 65°W, converts the results into ASCII format, and sends the URL of resulting data file back to the system. The system finally displays the query result as:

Date	trmm_retrieve(Granule,20,25,-70,-65)
1999-09-06	http://spring.scs.gmu.edu:8000/es/ 3B42.990906.5.HDF.subset.ascii
1999-09-07	http://spring.scs.gmu.edu:8000/es/ 3B42.990907.5.HDF.subset.ascii
1999-09-08	http://spring.scs.gmu.edu:8000/es/ 3B42.990908.5.HDF.subset.ascii

We are also exploring other mechanisms, such as eXtensible Markup Language (XML) based message passing, for the communication between data services and the system. This will make the system to easily interoperate with and integrate data services from systems using XML-based message passing for data service brokering service, such as the EOS ClearingHOuse (ECHO) system [11].

4. Prototype System

The prototypical system is a web-based system with the functionality of database loading, database browsing, data search, data retrieval, and data analysis [1]. We have conducted a series of experiments with example data services for TRMM rainfall data. In this section, we use some examples in support of hurricane research related data services to introduce the capabilities of the system. Kafatos et. al and Yang et. al. described a content-based online data search, access, and analysis system in [8,14].

TRMM is a mission to monitor tropical precipitation and to estimate its associated latent heating [2]. TRMM was launched on November 27, 1997 (EST) with 3 instruments, the Visible Infrared Scanner (VIRS), the TRMM Microwave Imager (TMI), and the Precipitation Radar (PR). TRMM/3B42 is a daily gridded rainfall dataset derived from the combined instruments on TRMM satellite. The geographic coverage is from latitude 40°S to 40°N and from longitude 180°W to 180°E. The spatial resolution is 1° longitude x 1° latitude. The file is in Hierarchical Data Format (HDF).

The system contains TRMM/3B42 data from January 1 of 1998, one data granule per day. Data_Granule table stores the records of Dataset_Id, Sequence, Date, West, East, North, and South Bounding Coordinate, and Granule for each data granule registered in the system. The Granule field records the filename and location of the data granules. The data granule itself resides outside of the system but is visible and accessible by the system.

Hurricanes are strong tropical cyclones in the North Atlantic Ocean. A hurricane has an organized convection (i.e. thunderstorm activity) and definite cyclonic surface wind and heavy rainfall. TRMM provides crucial observations from space to improve atmospheric modeling and data assimilation in the prediction of hurricanes.

We develop three data services, “rain_intensity” for data search, “trmm_retrieve” for data retrieval, and “storm_center” for data analysis. All of the three data services are developed with C and Fortran programming languages following the standard for data service development, and then compiled into executable processes.

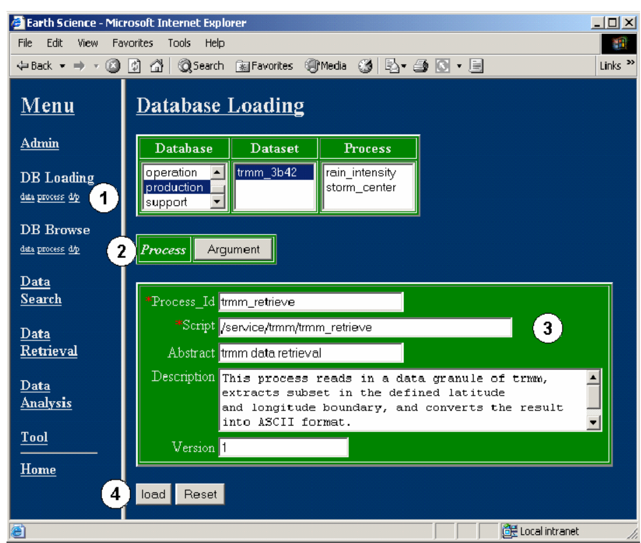

Figure 4. Database Loading

Database loading - Figure 4 shows the web user-interface and step by step procedure for data service providers to register the "trmm_retrieve" process into the system, as 1) from the menu frame, click on "process" link under database loading to open data service registration page, 2) click on "process" button in registration page to open the process registration form, 3) enter the information of process id, script, abstract, description, and version of “trmm_retrieve” process, and 4) click on "load" button to send the information of the process to the system.

The information entered in the script entry is full-path filename of “trmm_retrieve” process, which is visible and accessible by the system. Once receiving the information over the Internet, the system will ingest the information into the process table of the database.

After ingesting the information of process, users further provide the information of parameters of “trmm_retrieve” process, and the system store it in the argument table. With the same method, “rain_intensity” and “storm_center” data services can be registered into the system.

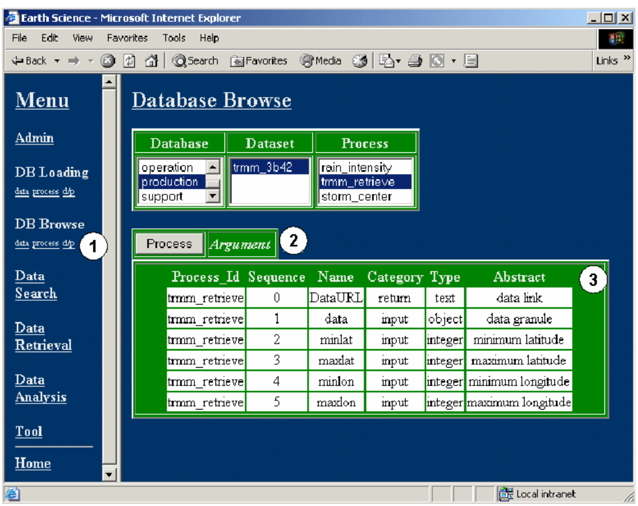

Figure 5. Database Browsing

Database browsing – The system enables users to browse the information stored in the database with database browsing function. Figure 5 shows the web page that displays the information of process id, sequence, name, category, type, and abstract for all the parameters of “trmm_retrive” process.

Data search - There are two TRMM data ordering systems. One is run by the Goddard Space Flight Center Earth Science Distributed Active Archive Center (GES-DAAC). The other one is run by the TRMM Science Data and Information System (TSDIS). The two TRMM data ordering systems provide metadata-based search by which users can search data by dataset, time, location, etc. For example, scientists know the temporal period of hurricane “Floyd” 1999 as 1999-09-10 to 1999-09-16 from the weather record. They can search the data by time.

But many scientists would like to search for data based on the attributes of physical phenomena and the values of physical parameters, such as severe rain events. For example, scientists would like to find out data of severe rain with rainfall rate greater than 4 mm/hour. Figure 6 shows the content-based data search with “rain_intensity” data service. From the data search page, users first enter the criteria of metadata-based search, dataset, location, and time. Then they enter the criteria for content-based search, rain_intensity(granule,0,40,-80,-40) > 4. Here (0°N, 40°N, 80°W, 40°W) represents the region of North Atlantic Ocean. After receiving the request, the system will construct a query statement like the following to search out the data from the database.

```
SELECT date, granule
FROM data_granule
WHERE dataset_id = 'trmm_3b42'
AND date >= '1999-09-10'
AND date <= '1999-09-16'
AND rain_intensity(granule,0,40,-80,-40) > 4
```

The system will find appropriate data of TRMM/3B42 dataset, covering the North Atlantic Ocean, and during the period of hurricane “Floyd”; and then invoke the “rain_intensity” data service process to examine the found data objects one by one to search out the data with rainfall rate greater than 4 mm/hour within the region of North Atlantic Ocean. The result shows that September 15 is a heavy rainy day satisfying the content-based searching criteria during the period of Hurricane “Floyd”.

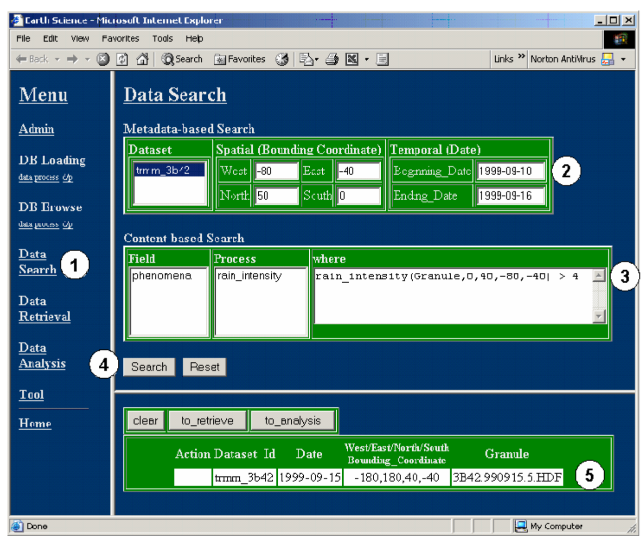

Figure 6. Data Search

In [9], Kelley et. al. described a system with capability of searching large-scale rain event of TRMM.

Data retrieval - Data retrieval is one of the most time and resource consuming activities involved in the research and application using Earth science data. To provide a data retrieval service, such as subsetting, resampling, and reformatting, data service providers usually need to develop a retrieval program with variety of languages and port into variety of operating systems considering the programming capability and computing environment of their users. Data service providers also need to write detailed documentation for their users to follow in order to build the retrieval program in their own computing environment. The users then need to download data and use the program to process the data into their desirable form. So any data retrieval service becomes a tedious programming effort for both data service providers and their users.

In our system, data service providers need to develop a retrieval program in only one favorable language in their own computing environment, and register it into the system as a data service. Their users can invoke the data service to retrieve the data into their desired form through the user interface of the system remotely. The users do not need programming at all.

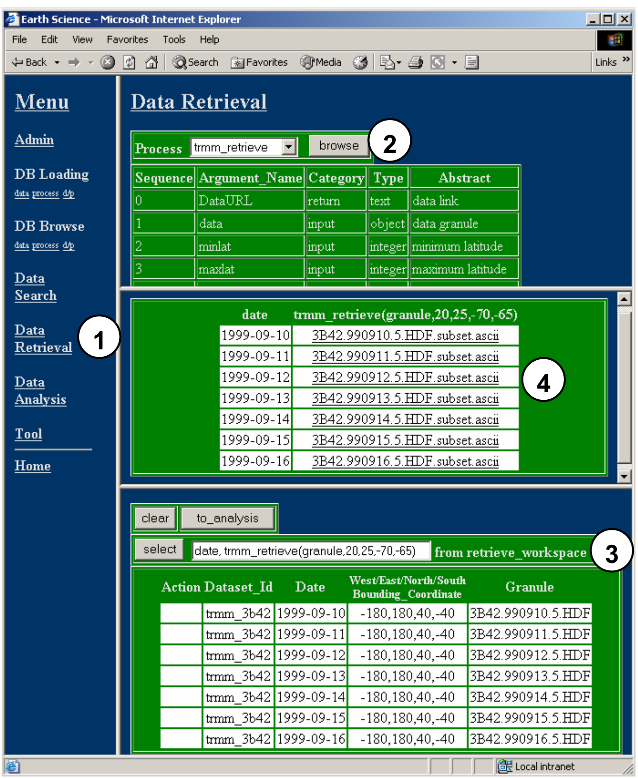

Figure 7a. Data Retrieval

Figure 7a shows the service-oriented data retrieval. From the data retrieval form at the bottom frame, users enter "trmm_retrieve(granule,20,25,-70,-65)" in the select query entry. The system will invoke the "trmm_retrieve" data retrieval service with the following query statement.

```
SELECT date, trmm_retrieve(granule,20,25,-70,-65)
FROM data_granule
WHERE dataset_id = 'trmm_3b42'
AND date >= '1999-09-10'
AND date <= '1999-09-16'
```

The "trmm_retrieval" process extracts the subset within the region of (20°N, 25°N, 70°W, 65°W), converts the result into ASCII format, and return the URLs of resulting data files. The system displays the results in the middle frame of data retrieval page. Users can click on the hyperlink link of resulting data, and the system will open a new web page to display the ASCII data (Figure 7b).

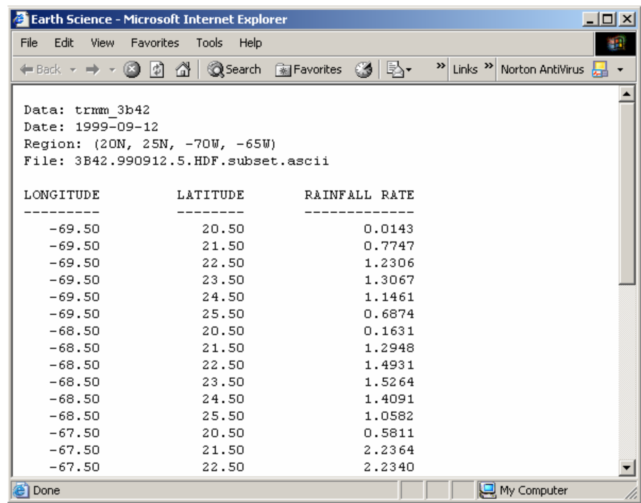

Figure 7b. Data Retrieval

Data analysis – A hurricane can be devastating with its strong winds and heavy rains. One of the important meteorological services is timely and accurate prediction of storm track or hurricane path. With traditional data systems, scientists need to transfer data to their local computing environment, and then perform the storm track analysis on the data. But they often do not have sufficient network bandwidth and disk space to transfer and store the massive amounts of data, therefore cannot analyze and/or predict the hurricane path timely.

Our system allows data service providers to register the storm track analysis process into the system as a data analysis service. Scientists can invoke the data analysis service to perform online analysis on the massive data from archive through the user interface of the system remotely.

Figure 8 shows the storm track analysis. Users enter "storm_center(granule,0,40,-80,-40)" in the data analysis form. Once receiving the request, the system will invoke the "storm_center" process from the following query statement.

```
SELECT date, storm_center(granule,0,40,-80,-40)
FROM data_granule
WHERE dataset_id = 'trmm_3b42'
AND date >= '1999-09-10'
AND date <= '1999-09-16'
```

The "storm_center" process analyzes the data granules one by one to find the storm centers during hurricane “Floyd” of 1999. The system displays the storm centers in the middle frame, forming the hurricane path. This system avoids the massive data migration, and enables scientists to analyze and/or predict hurricane path in time.

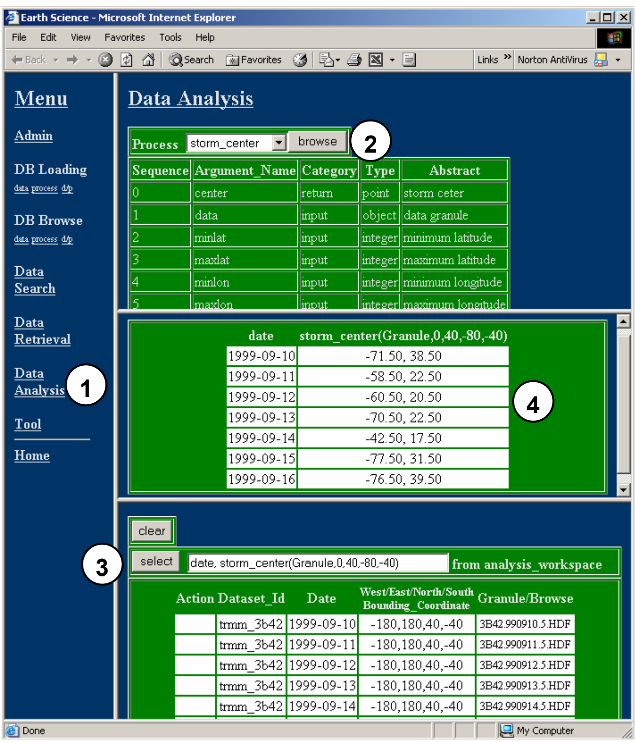

Figure 8. Data Analysis

In [10], Liu et. al. described a TRMM online analysis and visualization system supporting regional and time series analysis of rainfall.

5. Implementation Status

We have produced a prototype system. The prototype has a web-based user interface for the system to collect the information of data, process, and data processing from users, and to display the results to them. The users also invoke data services through the web user interface. Berkeley DB is used for the management of the underlying database. Berkeley DB is a light-weight embedded database management system. The prototype is a light-weight system with small footprint of 500 KB. The prototype runs behind a standard web server supporting CGI program, and does not need any additional software. Once started, the prototype system will work automatically without any need of administration and maintenance. The prototype system runs on both Unix and Windows Operating Systems.

The prototype system is functionally complete. We are developing dedicated systems based on the prototype system for supporting data service applications on TRMM rainfall data and on SeaWinds ocean surface wind data.

We are also developing the JDBC application in the database access component. Therefore the system can use any JDBC-compliant database management system for the management of underlying database of the system.

6. Conclusions and Further Developments

Data services become increasingly important in Earth science applications using large volumes of heterogeneous and distributed data. Existing data systems are lacking support of the data service oriented applications. In this paper, we introduce a system to support this class of applications. We first presented the system architecture, software components, data model, functionality, and user interfaces of the system. We followed with a prototype system to demonstrate the capabilities of the system using a series of data service scenario in data search, retrieval, and analysis of TRMM rainfall data.

In designing and development, we adopted the concept of object-relational technology. The system uses relational data model for the management of information of data, process, and data processing. The system supports the registration and utilization of data services in the system. In this system, we significantly increase its capabilities in dealing with the data services. We developed the standards for data service development that enables service providers to develop a data service with any kind of general programming language, similar to CGI programming in the web architecture. We also developed communication protocols to handle the interaction between the data services and the system.

This system extends the concept of user-defined functions in DBMSs into general data services in data systems, and makes the technology useful in a wider range of applications and usable by wider range of users. We are further developing the work in this extension.

In the current prototype, the system only handles the explicit return result of data service process. In many data service processes, the values of parameters are changed during data processing, and the parameters are used as implicit output. We will develop more sophisticated communication mechanism between data services and the system to handle the situation, for example, using the nested dot notation.

With data services becoming more and more important, more and more data systems are addressing this issue. For example, NASA EOS Data Gateway (EDG) is evolving to EOS ClearingHOuse (ECHO). ECHO will serve as a metadata clearinghouse like EDG and also serve as a service broker. To better serve its users, the Goddard Earth Science DAAC is developing a prototype TRMM Data Mining Service [3]. The technology we develop will be useful for the data service development of data systems, such as GES-DAAC. We are also further developing the system to easily interoperate with and integrate the data services from other systems, such as ECHO.

This system is very much a work in progress. It will evolve from the current prototype into a more powerful system supporting Earth science data services.

7. References

[1] http://spring.scs.gmu.edu:8000/es/index.html, "A Software to Support Earth Science Data Services", Project Web Site, Center for Earth Observing and Space Research, School of Computational Sciences, George Mason University.

[2] http://trmm.gsfc.nasa.gov, "Tropical Rainfall Measuring Mission", Project Web Site, NASA Goddard Space Flight Center, Greenbelt, Maryland.

[3] http://daac.gsfc.nasa.gov/CAMPAIGN_DOCS/hydrology/hd_datamin_intro.shtml, "Data Mining at the GES DAAC", NASA Goddard Space Flight Center, Greenbelt, Maryland.

[4] Brown, P. and M. Stonebraker, 1995, "BigSur: A system for the management of Earth science data", The 21st International Conference of Very Large Data Bases, Zurich, Switzerland, 1995.

[5] Chi, Y., C. Mechoso, K. Sklower, M. Stonebraker, R. Muntz, and E. Mesrobian, 1997a, "An output management system for an Earth System Model", Proceedings of the 13th International Conference on Interactive Information and Processing System (IIPS) for Meteorology, Oceanography, and Hydrology, pp. 409-412, February 2-7, 1997, Long Beach, California, American Meteorology Society.

[6] Chi, Y., C. Mechoso, M. Stonebraker, K. Sklower, R. Troy, R. Muntz, and E. Mesrobian, 1997b, "ESMDIS: Earth System Model Data Information System", Proceedings of the 9th International Conference of Scientific and Statistics Database Management, pp. 116-118, August 11-13, 1997, Olympia, Washington, IEEE Computer Society.

[7] Chi, Y., C. Mechoso, R. Troy, K. Sklower, and M. Stonebraker, 1998, "A data information system for an Earth System Model", 1998 Conference on Mission to Earth: Modeling and Simulation of the Earth System, pp. 111-114, January 11-14, 1998, San Diego, California.

[8] Kafatos, M., Z. Li, R. Yang, X. Wang, P, Hertz, D. King, H. Weir, H. Wolf, and D. Ziskin, 1997, "The virtual domain application data center: Serving interdisciplinary Earth scientists", Proceedings of the 9th International Conference of Scientific and Statistics Database Management, pp. 264-276, August 11-13, 1997, Olympia, Washington, IEEE Computer Society.

[9] Kelley, O., J. Stout, and M. Kafatos, 2000, "Content-based browsing of data from the Tropical Rainfall Measuring Mission (TRMM)", Proceedings of the 13th International Conference of Scientific and Statistics Database Management, pp. 270-273, August 15-17, 2001, Fairfax, Virginia, IEEE Computer Society.

[10] Liu, Z., L. Chiu, H. Rui, W. Teng, and G. Serifino, 2002, "Online analysis and visualization of TRMM and other precipitation data sets", The 18th International Conference on Interactive Information and Processing System (IIPS) for Meteorology, Oceanography, and Hydrology, January 2002, Orlando, Florida, American Meteorology Society.

[11] Wichmann, K. and R. Pfister, 2001, "ECHO - A message-based framework for metadata and service management", Proceedings of the NASA Earth Science Technologies Conference, June 11-13, 2002, Pasadena, California.

[12] Stonebraker, M, 1986, "Object management in Postgres using procedures", Proceedings of the 1986 International Workshop on Object-Oriented Database System, pp. 66-72, September, 1986, Pacific Grove, California, IEEE Computer Society.

[13] Stonebraker, M. and P. Brown, 1999, "Object-Relational DBMSs: tracking the next great wave", second edition, Morgan Kaufmann Publishers, San Francisco, California.

[14] Yang, R., C. Wang, M. Kafatos, X. Wang, T.A. El-Ghazawi, 1999, "Remote data access via SIESIP distributed information system", Proceedings of 11th International Conference on Scientific and Statistical Database Management, p. 284, July 28-30, 1999, Cleveland, Ohio, IEEE Computer Society.