Enabling User-Oriented Data Access in a Satellite Data Portal*

Rajesh Kalyanam, Lan Zhao, Taezoon Park, Larry Biehl, Carol X. Song
Rosen Center for Advanced Computing
Purdue University
West Lafayette, IN 47907

{rkalyana, lanzhao, tzpark, biehl, carolxsong}@purdue.edu

ABSTRACT

This paper presents the design and implementation of a web-enabled platform for enhancing user data access experience in a satellite data portal. The platform integrates a number of user-oriented capabilities including the subscription, production, dissemination, and visualization of custom satellite remote sensing data hosted at the Purdue Terrestrial Observatory (PTO). Different from a traditional remote sensing system, the PTO data portal aims at enabling access to real-time satellite remote sensing data products for a large, diverse user community. Central to this system is a user-driven publish/subscribe model that empowers users with the ability to specify, control, and receive satellite data products without having to go through the time-consuming and error-prone manual process of configuring the system to generate the requested data. The user-oriented capabilities are implemented on top of a service-oriented architecture (SOA) backed by an existing satellite data receiving and processing backend. To control the processing of user data subscription requests, a workflow is created by leveraging the SOA interfaces and mechanisms. The user-oriented platform also embraces the recent Web 2.0 technologies in bringing the rich access, sharing, and collaboration capabilities available in the general cyberspace to the scientific user communities.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Online Information Services – *Data sharing, Web-based services.* H.5.2 [**Information Interfaces and Presentation (e.g., HCI)**]: User Interfaces – *User-centered design.*

General Terms

Design

Keywords

Portal, Satellite Data, Web Service, User Interface, Publish/Subscribe.

Rajesh Kalyanam, Lan Zhao, Taezoon Park, Larry Biehl, Carol X. Song, Enabling User-Oriented Data Access in a Satellite Data Portal, International Workshop on Grid Computing Environments 2007, Nov. 2007, Reno, NV, USA. An electronic version of this document will be available at http://casci.rit.edu/proceedings/gce2007

1. INTRODUCTION

The Purdue Terrestrial Observatory is an interdisciplinary remote-sensing research facility [16] that includes several satellite ground stations capable of continuous ingestion of remote sensing data from the NOAA GOES-12, NASA MODIS Terra/Aqua, NOAA AVHRR, and Fengyun-1D MVISR sensor systems. It is one of the resources that Purdue contributes to the TeraGrid. A broad range of satellite data products for a number of land and atmosphere disciplines can be generated based on the raw remote sensing data. The many data products derived from the satellite observations describe features of the land, oceans and atmosphere. They can be used to study processes and trends on local or global scale for real-time environmental remote sensing research and applications. They are widely used in science and engineering fields such as atmosphere science and climatology, hydrology, agriculture, ecology, transportation and logistics, aviation and air traffic safety, and civil engineering.

Facing the wide spectrum of satellite remote sensing data products, it is highly desirable for users to be able to interact with the satellite data receiving facility, customize the data processing algorithm, dynamically control the production of custom data products, and access the data of interest in a timely fashion. Although techniques to enable each of the above capabilities exist, unfortunately there have been few efforts in integrating them into a portal platform for existing remote sensing systems. The limitations of existing system are further described as follows:

First, most existing satellite data systems only allow users to access a small set of pre-configured data products. It is practically impossible to generate all possible satellite data products in real time and make them available to the user community. Such a practice tends to be inefficient anyway as most users are only interested in a small subset of data products at a time. At PTO, up to 30GB of raw data are received and processed each day. The data from the MODIS sensor aboard the Terra and Aqua satellites can be processed into 14 different products. Each of the data products can contain up to 50 different variables. Meanwhile, when a user requests a new data product or a product with different parameter or output settings, the user typically needs to request the technical staff to reconfigure the system for the generation of the specific data product(s). This process incurs a high maintenance cost and yet is error-prone.

Second, since satellite data acquisition is a continuous process, a user of such data for his/her research is likely to request continuous updates to the data he/she subscribes to. Unfortunately, current systems do not provide any service of automatic data updates. The refresh and renewal of data is primarily initiated by users using a manual procedure without proactive notification and data transport from the system. It is also desirable that the user can access the data updates any time anywhere, such as from a mobile device when working away from their desks or in the fields.

To address these limitations, we have designed and developed a web-based user-oriented platform, called SATPro (satellite data production/delivery system), on top of the PTO data portal for satellite data subscription, production, and delivery. SATPro serves as an interface between end users and the data acquisition and processing backend connected to the satellite ground stations. SATPro changes the way data is served to the user, who otherwise has to pick whatever is available on a menu of existing products. SATPro aims at giving users the ability to specify the data products they want and have the back-end data production system generate and deliver the products in near real-time.

The key enabling component in SATPro is a publish/subscribe architecture implemented on top of a service-oriented workflow. A web interface allows researchers and teachers to interactively select and subscribe to specific data products of their interest for a certain period of time. During that period of time, the new data and data updates will be delivered in a way chosen by the user, including via RSS feeds to users with a PDA device. Finally, the user will be reminded of the expiration of the subscription and be provided with an option to renew his/her subscription. The system also integrates the satellite data with several easy-to-use on-line visualization tools for the users.

The rest of the paper is organized as follows: Section 2 describes the design of the SATPro platform. Section 3 describes in greater detail the design and implementation of the subscription mechanism. Section 4 describes the implementation of the portal interface as well as the integration of several on-line visualization tools. Section 5 discusses related work, and Section 6 concludes this paper.

2. SYSTEM DESIGN

The goal of SATPro is to create a user-oriented, interactive, and web-based subscription/production platform for delivery of the PTO satellite data. It allows users to not only select existing raw and derived remote sensing satellite data products, but also define new data products from a set of satellite data streams received by the ground station, and subscribe to the data products in a near-real-time manner. Behind the scene, SATPro automatically configures the data production back-end to perform the appropriate data processing algorithms on the raw satellite data, generate the specified products, and deliver them to the user in a way he/she chooses. In addition, it is most important to make the user interface easy to use for the domain scientists.

With such a goal in mind, we have designed the SATPro platform with a flexible user interface based on a service-oriented architecture and using Web 2.0 technologies [12]. It provides services through standard web service interfaces. A workflow is created to dynamically chain the service pipelines that enable/disable customized data generation and subscription. Externally the user-centered Web 2.0 enhanced portal interface provides several options for data delivery, online visualization tools, as well as subscription predicate sharing among users. Although open to interpretation, Web 2.0 represents a set of new technologies that drive a new generation of web applications. It brings services to the web with an emphasis on online collaboration and sharing among users. Several Web 2.0 features are used in the SATPro design: Tagging helps users contribute and share subscription predicates; JavaScript AJAX is used to create a rich, interactive, and user-friendly interface; RSS feed is used to push up-to-date satellite data products and updates to mobile users. More importantly, the entire system design is centered on strengthening the user's role and putting users in active control of the data products to be produced by the back-end data manufacturer.

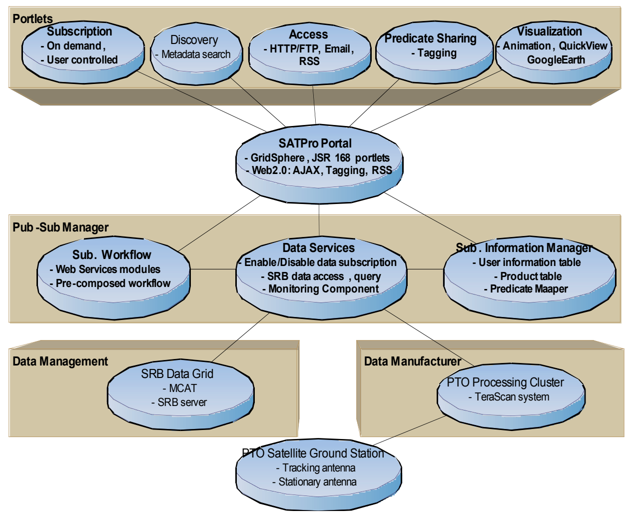

Figure 1. SATPro System Components

As shown in Figure 1, the SATPro system consists of the PTO data processing cluster, a Pub-Sub manager, a SRB (Storage Resource Broker) based data grid, and a web portal front end [1]. The PTO data processing cluster connects directly to the satellite receiver at the ground station. The Pub-Sub manager enables users to dynamically control the data production and delivery process at the processing cluster. The portal front-end is an easy-to-use interactive data access point implemented using standard JSR-168 compliant portlets and Web 2.0 technologies including AJAX, tagging, and RSS [8]. Historical data are managed using SRB data grid and can also be accessed through the portal. Details about the SRB data management system can be found in [9].

Since SATPro is centered on user-oriented data subscription and production, we will focus on this aspect of the system for the rest of the paper.

3. ON-DEMAND DATA SUBSCRIPTION AND PRODUCTION

Figure 2 illustrates our user-driven publish/subscribe model for data production and processing. The underlying publish/subscribe design decouples data providers and data consumers, leading to a loosely coupled architecture with better flexibility and scalability. On top of this model, a user feedback loop is created via the Pub-Sub manager. It provides the data consumer fine-granularity control over the data manufacturer on which products to generate and for how long. As a result, the system is more resource-efficient and adaptive to the user's need. Users enter subscription details through a web interface and the Pub-Sub manager sends updates to users when new data are available. Within the Pub-Sub manager, a MySQL database is used to store the subscription information for each user and to maintain the dynamic list of users subscribing to a particular product. The core of the Pub-Sub manager is a monitoring agent component which triggers the notification process for each data subscriber when new products are generated.

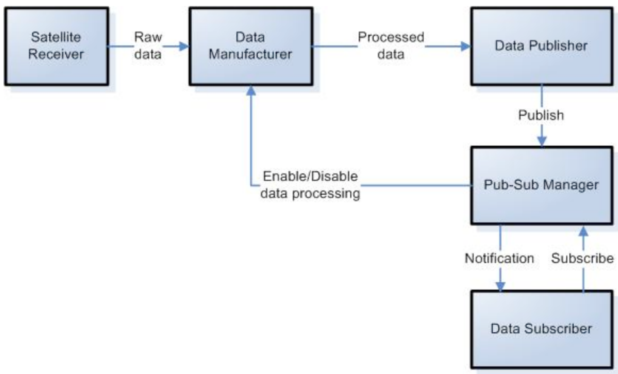

Figure 2. A User-driven Publish Subscribe Model

While most of the aforementioned components form the consumer side of the subscription process, the producer side is responsible for data production and publishing to a web repository for future download. The subscription process leverages the existing data manufacturer components and adds a signaling layer which controls the start and ending of data production. It also adds a remote copy mechanism that moves the data products to the web repository where users can download the products. This implementation of the publish/subscribe model is different from the typical model based on standards such as the JMS (Java Message Service) [7]. JMS provides a highly flexible and efficient means of asynchronous communication between various components of a distributed system. The messages exchanged between these components can range from simple text to complex objects. In addition the purpose of message exchange can be to simulate a complex interaction mechanism. Our requirements in the subscription system, however, are much simpler. We have a central manager that receives notifications when new data has been generated. The role of the manager is to send email notifications to users subscribed to this data product. These messages typically include a simple link to the image files and the users do not perform any complex interaction in response to the notifications. Although we do not require the complete flexibility or complexity of current messaging standards, it might be instructive to compare the relative performance of messaging models based on a hierarchy of topics for representing the data products with our current database query based approach.

3.1 Subscription Interface

The SATPro subscription interface supports a flexible product specification schema for users to specify data subscription information. The schema currently supports the specification of fine-grain products at the product level and selected ones at the variable level such as data for specific channels, RGB images (in MOD02), the vegetation indices NDVI and EVI (in MOD09), and Corrected Optical Depth Land (in MOD04). Users can further narrow down their specification along multiple dimensions, including the geographic study area and the data collection time periods. Users also have the choice of creating output products in 17 different map projections and 10 different file formats. They can further specify composite images that combine data collected across several days to reduce the impact of clouds in the image data.

The subscription interface presents a series of choice lists corresponding to various data products and their variants. Due to the sheer number of possible combinations, we make use of the JavaScript AJAX technique to filter each choice list based on the selections made in the preceding choice lists so that the user can only make a valid product selection. The selected product specification is then translated into a subscription predicate which has a highly structured form consisting of filters, usually in the form of name-value pairs of properties and comparison parameters. Each filter defines a constraint and can be logically combined to form complex subscription predicates. Each product can ultimately be broken down into several constraint fields such as the satellite source (e.g. AQUA or TERRA), the region of coverage (e.g. Indiana or all pass), the product type (e.g. MOD01-MOD35, Level1, Turbidity, NDVI, Cloud Mask, Visibility and Sea Surface Temperature), the projection type (e.g. UTM or ORTHOGRAPHIC) and the data format (e.g. JPG, GeoTIFF or HDF).

However this user interface is intuitive only for users who are familiar with the mapping from their desired products to these constraint fields, especially with the list of product types. Typically users have only an informal description of the data products such as "*a bimonthly composite image of vegetation data*" or "*a Level 1B MODIS image obtained from the first 6 channels with a sun elevation angle greater than 20 degrees*". This discrepancy arises mainly from the gap between the end-user's perception of the data and the perception of the system developer. To also handle such product descriptions in the subscription request, we are augmenting the web interface to provide an additional method of subscription specification using a limited subset of plain English, which is then translated into a subscription predicate by a *predicate mapper*. It is important to note here that due to ambiguity in the language, a given description may map to several product definitions. We handle this ambiguity by providing users with a list of product definitions that match their input description, with a thumbnail of the most recent image from that product. Users can then select the product that they are interested in or revise their product description.

The predicate mapper performs the matching by searching for keywords relevant to each product description in the user's specification. Each product description contains an initial set of keywords that could possibly refer to the data from this product. For example, "*vegetation*" is a keyword that refers to the data obtained from the NDVI measurement processing. We calculate a score for each (keyword, product) pair based on the frequency of usage of this keyword while selecting this particular product. Given a new user product description, we calculate the total score for each product based on the scores defined above and the keywords in the description. Those products with the top three scores are then presented to the user. The user selection in turn helps us refine the initial set of keywords and thus improve the mapping of descriptions to products based on the input from multiple users. Keywords that are not already in our initial list are then added based on their frequency of usage by several users. This calculation can be expressed formally below:

Suppose we have a table T of scores for each <product, keyword> pair. The score s_{ij} for a given product p_i and keyword k_j is given by T(i,j). Given an English description of the product by a user that contains the keywords $k_1,....,k_n$, the cumulative score for each product p_i would then be $S_i = s_{i1} + s_{i2} + ... + s_{in}$. We then choose the top three products based on the sorted list of S_i values. Precise matching of user keywords to products is a complex problem. Our approach is pragmatic, starting with this very simple scoring scheme, and with user assistance, we can then identify relevant products. Our score based method above is based on a restricted set of products and is mainly used to highlight the design of our pseudo-learning based subscription system. To this end, we propose a simple frequency based semantic processing method for identifying subscription products. It will benefit from some ontology based semantic processing methodologies using a formal set of representation which includes a vocabulary for referring to terms in the subject and the logical statements that describe what the terms are and how they are related to each other in the ontology. There are a few projects whose focus is to develop an ontology specific to addressing applications in the remote sensing domain [19]. We are yet to assess the performance of different methods and develop an integrated solution based on ontology for remote sensing data.

Each of these data products has an additional set of variables that place further constraints on the data processing and the results returned. Once a data product has been selected on the web interface, we allow users to specify these additional constraints on the variables of that product. This is accomplished by allowing users to dynamically combine conditions by logical OR and AND operators. Each condition involves a choice of a variable, a comparison operator (equals, less than, greater than, between) and a value to compare against. The concatenation of these individual conditions is then stored along with each user subscription on our system. The web interface also provides for an input of the subscription start date and the number of days to be subscribed for a product. These two inputs are needed to provide the user with an option of setting up a temporal subscription. On the expiration of the subscription a notification is sent to the user to renew the subscription request if they want to. To make the portal more customized to the interests of each user, it records the last data product the user subscribes to and automatically fills up the related fields the next time the user loads the subscription portlet.

As a by-product of the simple scoring scheme mentioned earlier, and combined with the popular Web 2.0 tagging technique, we can enable the sharing of the subscription predicates. When a user successfully defines the target product, the subscription predicate constructed will be tagged and catalogued in the system. This predicate catalog will allow other users to quickly zero in on the products of their interests by providing a few keywords, which are used to generate a list of products whose tags match. As the subscription community grows, the catalog will contain an increasing number of the subscription predicates and thus helping the users with an easier subscription process.

3.2 Subscription Information Management

The subscription information system connects the front-end interface with the Pub-Sub manager to handle subscription requests. User subscription predicates are stored in a table in a central MySQL database and will be evaluated when a new data file is received. The result will identify the list of users who have a valid subscription for this product and an email would be sent to the address stored in the subscription record, with details on how to obtain the data file.

As mentioned earlier, the core of the SATPro subscription system is a monitoring component that uses the subscription table to find the set of users subscribed to a product when new data are available. More specifically, the monitoring component steps through the subscription records and tests the predicates against the value of the variables for the new data file. We maintain a list of product keys and a function from each data file to a tuple of key values. Based on the key values of a new data file, we can evaluate each subscription predicate in the subscription predicate table to determine if the corresponding subscriber needs to be notified. The data products also constitute a foreign key in the subscription table. A simple product name scheme is designed which strings together the set of constraints that make a product predicate including fields such as the satellite name, product name, projection type, coverage area and file format. This facilitates easy insertion of the product names in the subscription table. In addition the product information table stores a frequency value which indicates an approximate production frequency for this product which in turn depends on how frequently new data is received from the satellites. Information collected here can be used to enhance service and production of the most demanded products.

We include below snippets of the two database table schema, at a minimum level of implementation, in XML format,. The *ProductInfo* table stores information about each of the data products, with specific reference to their storage path on the web repository and the filename patterns for the data files.

```
<PRODUCTINFO>
 <Field>
     <Name>product</Name>
     <Type>varchar(40)</Type>
     <Null>YES</Null>
     <Key></Key>
     <Default>NULL</Default>
 </Field>
 <Field>
     <Name>filepattern</Name>
     <Type>varchar(40)</Type>
     <Null>YES</Null>
     <Key></Key>
     <Default>NULL</Default>
 </Field>
 <Field>
     <Name>path</Name>
     <Type>varchar(80)</Type>
     <Null>YES</Null>
     <Key></Key>
     <Default>NULL</Default>
 </Field>
 <Field>
     <Name>age</Name>
     <Type>int(11)</Type>
     <Null>YES</Null>
     <Key></Key>
     <Default>NULL</Default>
 </Field>
…
```

```
</PRODUCTINFO>
```

The *PtoSubscribe* table stores subscription information for all users.

```
<PTOSUBSCRIBE>
  <Field>
      <Name>username</Name>
      <Type>varchar(80)</Type>
      <Null>YES</Null>
      <Key></Key>
      <Default>NULL</Default>
  </Field>
  <Field>
      <Name>predicate</Name>
      <Type>varchar(300)</Type>
      <Null>YES</Null>
      <Key></Key>
      <Default>NULL</Default>
  </Field>
  <Field>
      <Name>product</Name>
      <Type>varchar(40)</Type>
      <Null>YES</Null>
      <Key></Key>
      <Default>NULL</Default>
  </Field>
  <Field>
      <Name>start_date</Name>
      <Type>date</Type>
      <Null>YES</Null>
      <Key></Key>
      <Default>NULL</Default>
  </Field>
  <Field>
      <Name>end_date</Name>
      <Type>date</Type>
      <Null>YES</Null>
      <Key></Key>
      <Default>NULL</Default>
  </Field>
  <Field>
      <Name>email</Name>
      <Type>varchar(80)</Type>
      <Null>YES</Null>
      <Key></Key>
      <Default>NULL</Default>
  </Field>
</PTOSUBSCRIBE>
```

Based on the table schema above, the following statement retrieves the email addresses of users who have valid subscriptions to a given product:

```
SELECT  email, end_date
FROM  ptosubscribe
WHERE   predicate == TRUE for product AND    curdate()
BETWEEN start_date and end_date;
```

3.3 Data Manufacturer Process

The SATPro subscription system leverages an existing data production system responsible for processing the raw streaming satellite data and converting it to advanced data products. The PTO data processing system runs a proprietary TeraScan software system from SeaSpace [18]. It has a daemon process that monitors the incoming satellite data stream, scans the product configuration files, and creates the derived data products using a 10-node Linux cluster managed by PBS batching system. Each time a user specifies a new data product, a configuration file is created by the subscription information system which will be automatically processed whenever new data arrives from the satellite receiver.

3.4 Monitoring Component

Since satellite data production is a continuous process, we need a means of detecting new updates and sending notifications to subscribed users.

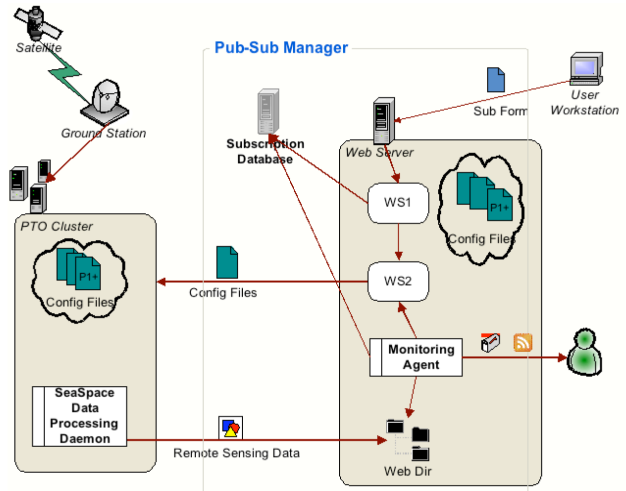

Figure 3. Pull-based Data Subscription

Our first solution is to implement the monitoring agent component on the client side where we have more control. The only requirement of this approach is that we transfer the output to a location where it can be monitored by the agent. This is not a very strict requirement, since we would eventually need to transfer the data to a user-accessible web server from which it can be served. This method forms the "*pull*" mode of monitoring as shown in Figure 3.

The client-side monitoring agent consists of a Perl script that first queries the subscription table for a list of products that users are currently subscribed to. Next it queries the local repository path to monitor for each of these products to look out for. The script then checks for any new data files and sends a notification with the file path to all users subscribing to this product. Once the script has completed a cycle through all possible products, it stays inactive for a certain period of time. This is because data generation is not very frequent and depends on when the satellite performs the same sweep again. The frequency field in the product information table is used to store this information. The highest frequency of data production can be determined as the minimum among all the values using a query statement "select min(age) from productinfo;". The monitoring agent then sleeps for this period of time and runs another cycle after that. New files could be automatically determined as files that are "fresher" than this time period, since they would not have been picked up in the previous cycle of the script.

Although the pull-based approach is easy to implement, it has the drawback of delay in data delivery and extra traffic caused by checking the local data repository and query against the subscription database. To improve the responsiveness of the system, we further designed and implemented a "push" mode solution where the data manufacturer is enhanced to be responsible for providing notifications on availability of new data. There are two main challenges in implementing this approach: It requires close integration with the TeraScan data processing package running on the PTO cluster and it is difficult to modify the existing process to add the notification step due to the nature of the implementing process. In addition, invoking any additional code on the data manufacturer server implicitly assumes the availability of an execution platform and the necessary libraries.

To solve these aforementioned challenges, we implemented a modular solution as illustrated in Figure 4. It is based on several web services modules connected through the JOpera workflow runtime engine [4]. There are three web services connected in the data subscription pipeline running on the client side:

- *WS1*: upon receiving a user request, it enters the information into the subscription database;
- *WS2*: based on the user request, it generates and copies the corresponding data configuration files to the head node of PTO cluster;
- *WS3*: each time a new data product is generated, it is invoked by a monitoring agent running on the head node of PTO cluster to check the subscription database and then send out notification.

As a result, the steps are modularized and the workload on the data manufacturer server is reduced. The monitoring agent is a shell script that invokes *WS3* through a Java client. It leverages TeraScan's support of custom binary execution at the end of a data product configuration to ensure real-time update notification.

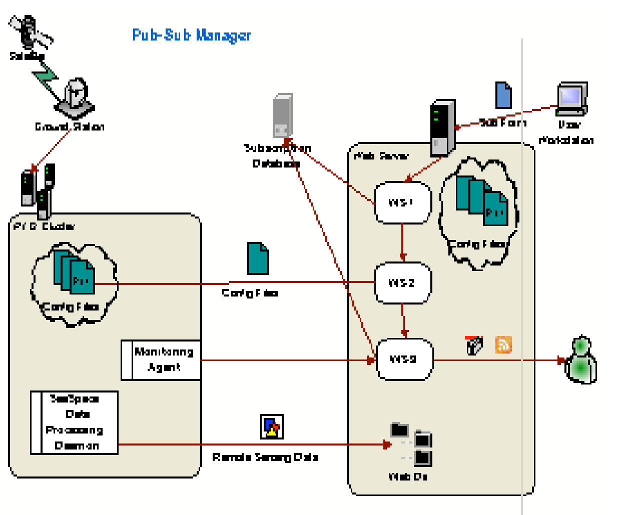

Figure 4. Push-based Data Subscription

3.5 Data Delivery

There are several modes of data delivery when new data is available: users get real-time notification via email and can download the data from a web server. Older data on the web server will be regularly deleted after a certain period of time has passed since its generation. Since the data is served in a read-only fashion from the web server, there is no need to further secure the data or implement concurrency access control. We are also in the process of enabling "push" mode delivery options: the data can be automatically pushed to the user's site via FTP. In addition, the data is preprocessed to generate a thumbnail image which can be pushed to the user via RSS feed. Because of the compact size of the image, researchers in the field are able to view them in real time using their handy mobile devices.

4. PORTAL INTERFACE

The portal interface serves as a single access point for several common functionalities in conjunction of the PTO remote sensing data. It was developed using the GridSphere portal framework [6]. The GridSphere framework provides an open-source portlet API that is JSR-168 compliant, a flexible XML based presentation description that allows customization of the visual interface, as well as a portlet service model for encapsulating, reusing, and sharing portlet logic. Currently the portal supports the following functionalities: data browse, query, subscription, download, and visualization.

To make the interface easy to use for the domain users**,** they are provided with intuitive interfaces for data discovery and access. As shown in Figure 5, each of the data components occupies a tab in the data portal and these tabs further contain the various supported functionality in the form of sub-tabs. The portal also provides several on-line visualization tools including live animation of the latest satellite images, image overlay on Google Earth, and thumbnail quick view. Two screenshots are shown in Figures 6 and 7.

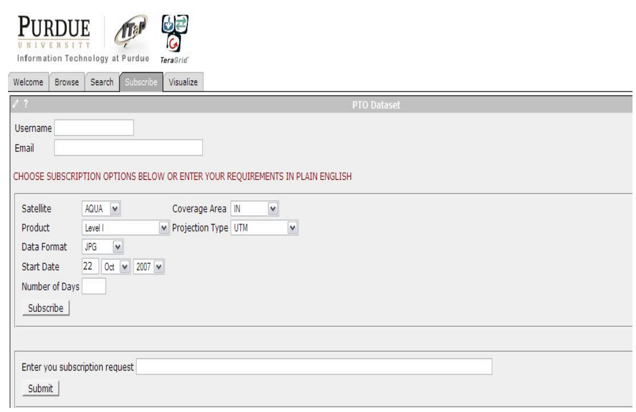

Figure 5. Portal User Interface

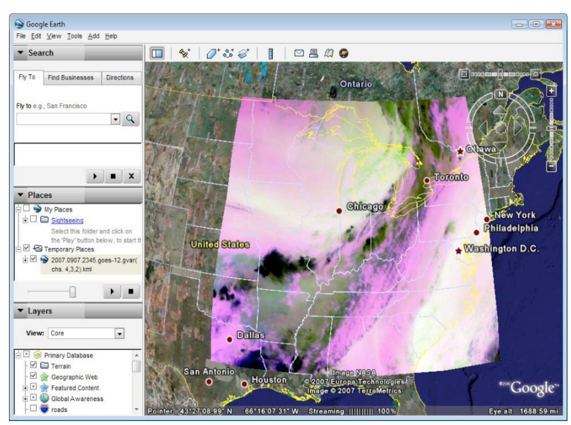

Figure 6. Overlay a PTO GOES-12 Image on Google Earth

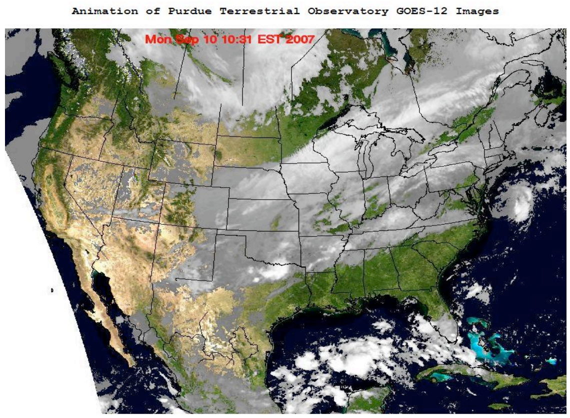

Figure 7. Animation of the latest GOES-12 Images Covering the US Continent

5. RELATED WORK

There are many existing science gateways that enable integrated data and/or application access in different domains. Typical science gateways provide users with a means of accessing scientific data and running post-processing operations on the data. The result data can be used for a wide array of prediction tasks for research and educational purposes. All these systems seek to provide an intuitive means of accessing data and scientific applications through a web interface. Several known examples include the LEAD (Linked Environments for Atmospheric Discovery) portal, NEES (Network for Earthquake Engineering Simulation), nanoHUB, GEON (GEOscience Network), QuickSim, and ESG (Earth System Grid) [5, 10, 13, 14, 15, 20].

The common theme among all these portals is the use of the computational resources to run resource intensive tasks and provide users with a friendly and intuitive interface to the output of the computational tasks. In addition users can also perform other data access tasks such as search and browse for existing data. Each of the science gateways differ in the specific scientific domain that they cater to. The LEAD portal provides users with a simulation system that makes use of the large computational resources provided by the TeraGrid framework to run a simulation on atmospheric and climate data streamed from different satellite sensors. The NEES portal, on the other hand, provides researchers with the simulation and data discovery tools to remotely access and collaborate on earthquake simulation data. The NCN (Network for Computational Nanotechnology) framework has the twin purpose of providing a computational framework for driving nanotechnology research while also providing an educational tool which can be used by students to run simulation runs as a part of their regular coursework. The nanoHUB provides a toolkit for users to put a graphical user interface in their applications so the tools can be made available on nanohub.org to the general community. Other science gateways cater to the domains of astronomical sciences, biological sciences, chemistry and materials research.

Our SATPro system also follows the general pattern for science gateways, providing a means of accessing and querying the remote sensing data acquired by different satellite sensors. In addition we provide a subscription mechanism which enables regular users to proactively specify, produce, and access new data products in a timely fashion. We believe this is the next logical level of capabilities that a scientific gateway should provide on top of the basic data access functionalities. Moreover, our system embraces the ongoing trend of applying Web 2.0 concepts and techniques in delivering web applications to end users. It creates an open, interactive, user-oriented environment where users have more control on the web application, contribute and share knowledge via tagging, and access near-real-time data updates even when working away from their desks.

Among the current satellite data portals, the MODIS portal from the Oregon State University direct broadcast station also supports data subscription [11]. After examining the limited documentation available about this site, it seems that they only support data subscription of MODIS data. It offers a limited set of products with a pre-defined data format, covering region as well as projection system. These data products are always being generated no matter whether there are uses subscribing to them. Our system instead allows the user the full freedom in defining the data product to be generated. Only those products with a valid subscription will be produced by the system. It is more scalable and is used on data streams from multiple satellites. It is also more flexible to accommodate various types of user requirements. Other satellite data portals/management systems such as the NOAA Satellite and Information Service (http://www.ncdc.noaa.gov/oa/ncdc.html), the MODIS portal at Space Science and Engineering Center (SSEC) (http://www.ssec.wisc.edu/data/), the CSR (Center for Space Research) portal (http://magic.tacc.utexas.edu/station/products/index.php), the ESL (Earth Scan Laboratory) portal (http://www.esl.lsu.edu/imagery/) only support browsing and searching for various satellite data. Some of them offer animation on the latest data as well.

6. CONCLUSIONS

We have described the design and implementation of a user-oriented subscription system which is integrated with an existing data generation process to provide users with the capability of initializing new satellite data production and receiving real-time automatic updates. A publish subscribe system is developed using web service based workflow technology. It is scalable to support multiple satellite data streams. It also offers flexibility to users in selecting the products that they are interested in, while improving the performance of the generation process by turning off the generation when no active subscriptions exist. The web portal enables interactive selection, generation, and delivery of the PTO satellite data products that are customized to match the interests of individual users. We do not currently provide any detailed performance study results since our focus in this paper has been the design of the system and its applicability to a real-world satellite subscription requirement. We hope to perform detailed studies of the performance based on the several factors such as the number of users and products provided and present them in some future paper. As a TeraGrid data provider, this new PTO subscription/delivery platform will encourage broader access from the grid user community.

7. REFERENCES

[1] Baru, C., Moore, R., Rajasekar, A., and Wan, M. 1998. The SDSC Storage Resource Broker. Proc. CASCON'98 Conference, 1998.

[2] Content Standard for Digital Geospatial Metadata" (CSDGM) Version 2 (FGDC-STD-001-1998), http://www.fgdc.gov/standards/documents/standards/metadata/v2_0698.pdf.

[3] Content Standard for Digital Geospatial Metadata: Extensions for Remote Sensing Metadata (FGDC-STD-012-2002), http://www.fgdc.gov/standards/documents/standards/remote_sensing/MetadataRemoteSensingExtens.pdf.

[4] Pautasso, C. 2005. JOpera: An Agile Environment for Web Service Composition with Visual Unit Testing and Refactoring. VL/HCC 2005.

[5] Earth System Grid (ESG), http://www.earthsystemgrid.org/.

[6] Novotny, J., Russell, M., and Wehrens, O. 2004. GridSphere: An Advanced Portal Framework. EUROMICRO 2004, 412-419.

[7] Java Message Service, http://java.sun.com/products/jms.

[8] JSR 168: Portlet Specification http://www.jcp.org/jsr/detail/168.jsp.

[9] Zhao, L., Park, T., Kalyanam, R., and Goasguen, S. 2006. Purdue Multidisciplinary Data Management Framework Using SRB. SRB Workshop, Vol. 1, pp. 6-11, February 2006.

[10] LEAD Portal, http://lead.ou.edu.

[11] MODIS portal from the Oregon State University direct broadcast station, http://sugar.coas.oregonstate.edu/MODIS/.

[12] Pierce, M. E., Fox, G. C., Yuan, H., and Deng, Y. 2006. Cyberinfrastructure and Web 2.0. Proceedings of HPC2006, July 4 2006, Cetraro Italy.

[13] Pierce, M. E., Fox, G. C., Aktas, M. S., Aydin, G., Gadgil, H., Qi, Z., and Sayar, A. The QuakeSim Project: Web Services for Managing Geophysical Data and Applications. PAGEOPH Special Issue for 5th ACES International Workshop, Island of Maui, Hawaii.

[14] nanoHUB, http://www.nanohub.org.

[15] NEES portal, http://neesforge.nees.org/projects/simportal/.

[16] Purdue Terrestrial Observatory, http://www.itap.purdue.edu/pto/.

[17] Kalyanam, R., Zhao, L., Park, T., and Goasguen, S. 2007. A Service-Enabled Distributed Workflow System for Scientific Data Processing. Proceedings of IEEE Int'l Workshop on Future Trends of Distributed Computing Systems (FTDCS'07), Sedona, AZ, March, 2007.

[18] SeaSpace Corporation, http://www.seaspace.com.

[19] Ram, S., Venkatsubramanyan, S., Yool, S., Marsh, S. and Park, J. 1998. SAGE: An Intelligent Agent Based Toolkit for Discovering and Accessing Distributed Earth Observing Data. Proceedings of the Earth Observation and Geo-Spatial Web and Internet Workshop 98, 1998

[20] Nambiar, U., Ludaescher, B., Lin, K., and Baru, C. 2006. The GEON portal: accelerating knowledge discovery in the geosciences. Workshop On Web Information And Data Management Archive, Proceedings of the eighth ACM international workshop on Web information and data management, 2006.