Abstract
Demand for high resolution data for various geospatial applications is increasing day by day. Need arises to procure data from foreign vendors to meet user needs and expand the capacity of data supply. National remote sensing center (NRSC) being the nodal agency, has the responsibility to procure the data from different foreign vendors on behalf of Indian users. Streamlining and automating the process workflow to enable easy ordering and monitoring the order status to enhance the user ordering experience is vital. Communicating the workflow tasks at regular intervals through instant messaging is the demand of the day. Integrating into the existing multi mission workflow is essential to improve resource reuse, performance and turnaround time. Difference in the order processing of IRS data and foreign high resolution data arises due to heterogeneous and foreign nature of data processing. User-friendly interactive web based online services become need of the hour to cater user demands. This paper describes the design and implementation aspects of foreign satellite high resolution data ordering system developed using oracle database for data repository and RIA technologies for user friendly interactive interface. It provides interlinking of vendor, user and processes to ensure continuity.

1. Introduction
As need for high resolution satellite data for various geospatial applications is increasing, apart from supplying data from Indian remote sensing satellite constellation demand for supplying data from third party vendors has become necessary. A user who wants this data has to contact NRSC Data Center (NDC) and the NDC takes the responsibility of processing the orders by contacting the vendor and the user at various levels of the order processing. As the existing processes flow is manual, automating the flow and enabling suitable handshake between the user, vendor and the processing manager has become vital. Designing processing states which drives the workflow and enabling user, vendor and processing manager to take appropriate action is a challenging task. As the data is acquired from third party vendor fitting the processing in sync with IRS data processing is also a major task. For IRS data processing the user requests are resulted into job orders which are ingested into production workflow comprising data product generation, value added services and quality control [6]. Therefore incorporating the foreign satellite data requests into job orders in compliance with IRS job order and ingesting them into existing production flow and customized the flow suitable to this unique need is the challenge to accomplish.

This paper describes the design, development and deployment of the web application which is called as the foreign satellite data order processing system.

Key features of the application are

- Ease of use, navigation and high response
- Customized vendor forms
- Communication history
- Instant messaging as required
- Handshake between user-vendor-processing managers
- Report generation
- Status based monitoring
- Event driven workflow
The system is designed to handle the functions of the user and processing manager. The user functions are available over the internet which includes placing data orders and monitor it. The processing manager functions are available over the intranet which includes validation, verification and processing of orders. To automate the workflow background processes and event based messaging functions are incorporated.

The internet accessible functions are
- Enquiry generation
- Feasibility download and confirmation
- Proforma Invoice(PI) confirmation
- Order confirmation(OC) by user
- Information upload and download
- Status monitoring
- Help for guidance

The intranet functions include
- Enquiry Validation
- Interaction with vendor and user
- PI and order generation
- Credit confirmation

2. Architecture

Foreign satellite data ordering system is designed using three layered architecture. Three-tier architecture is a client–server software architecture pattern in which the user interface (presentation), functional process logic ("business rules"), computer data storage and data access are developed and maintained as independent modules, most often on separate platforms[1].

2.1 Presentation layer

Presentation layer enables the user to enquire about the high resolution foreign satellite data and drives him to take status based actions. The layer is designed to register user interactions and pass information and events to the service layer. User can monitor the status of his orders and send additional information as required. The presentation layer guides users, processing manager and vendors to take appropriate actions. This rich graphical user interface is designed using adobe flex, action script, mate flex framework. The presentation is depicted in figure2.

2.2 Service Layer

Service layer incorporates system business logic. It receives information and events from presentation layer and process it. Service layer interacts with the presentation layer using BlazeDS. Plain old java objects (POJO) are being used to perform the business logic. Results of the business logic are returned to the presentation layer in appropriate framed objects. Service layer interacts with the data repository for fetching and manipulating the data. Service layer contains business logic and event processes. The service layer is depicted in figure3.
2.3 Repository

Repository contains diversified data comprising of user profiles, enquiries, orders, proforma invoices, email communications and payment details. The user profiles contain information about the registered users for the services which are organized based on the different categories. The enquiry and order collection database has information about user orders details. Order details consist of user enquiry details, feasibility details, proforma invoice details and purchase order details. Communication information comprises of email history of the handshake among user, vendor and processing manager. Payment related information is stored under billing and account details. Product and price details hold all the satellite, sensor, different types of products and price information. The repository is depicted in figure 4.

3. Design

3.1 Presentation Layer

The presentation layer is responsible for formatting and delivery of information to the application layer for further processing and displays [3]. It relieves the application layer of concerns regarding syntactical differences in data representation within the end-user systems. The presentation layer was designed to provide a rich graphical user interface which encompasses user friendliness and smooth navigation.

A User Interface is designed for ordering foreign vendor satellite data, monitoring of ordering status and sending custom emails by the user. Processing managers Interface is designed for administrative and control functions and to interact with the user and vendor. Presentation layer was implemented using MVC design pattern and mate flex framework. Mate is tag-based, event driven flex framework. This framework has been used to make it easy to handle the events that presentation layers create. It allows defining who is handling those events; whether data needs to be retrieved from the server or other events need to be triggered. In addition, Mate provides a mechanism for dependency injection to make it easy for the different parts of the application to get the data and objects it needs [2].

The presentation layer is internally segregated and customized based on user profiles such as users, processing managers and administrators. The graphical user interface is customized to manage the information and workflow guiding user-vendor-processing manager to take appropriate actions. The layout is managed within single page which leads to minimal navigation and user friendliness

3.2 Service layer

A Service Layer defines an application's boundary and its set of available operations from the perspective of interfacing client layers. It encapsulates the application's business logic, controlling transactions and coordinating responses in the implementation of its operations [4].

Business logic implements workflows and rules which are driven by events from the presentation layer. It establishes database connections using object pool design which leads to efficient use of database connections. To debug the applications activities and database, logs are created using singleton pattern. Session timeout is maintained to enhance security and resource efficiency.

Plain old Java object (POJO) contains the business logic and rules. A POJO is an ordinary Java object, not bound by any special restriction. This layer uses Adobe BlazeDS which is a server-based java remoting and web messaging technology that enables developers to easily connect to
back-end distributed data and push data in real-time to Adobe Flex applications for more responsive rich Internet application (RIA) experiences [5].

Service layer also contains event processing which coordinates between presentation layer and repository using BlazeDS. Event processor executes the workflows where process inputs and queries are received from presentation layer and response is rendered back to the presentation layer.

3.3 Repository

The repository stores all the application data on the server. It stores user profiles, order details, communication information, billing, account details, product, price details and work order Extensible Markup Language (XML) files. Information is structured into database tables. Primary key and foreign keys are defined for maintaining integrity and dependency of data. To remove redundancy and ensure data consistency 3rd form of normalization is used. History management of user orders is achieved using relational views which also improve performance. Order status and other statistical reports are derived using join queries to fetch data from multiple tables. XML is used to organize work order details which are sent to production workflow for masking and quality control.

4. Workflow

The workflow categorized into two types, one is user action workflow and the other one is processing manager action workflow.

4.1 User Action

User Action workflow enables user to select vendor form and place enquiry for high resolution data. It also enables users to download and approve vendor feasibility.

4.2 Processing Manager (PM) Action

Processing manager workflow enables manager to perform administrative and control functions like enquiry validation, upload feasibility, generate proforma invoice, upload order confirmation report (OCR), credit check, generate purchase orders and send the work orders to production for masking and quality control.

Figure 5: Workflow of user enquiry generation

Figure 6: Workflow of feasibility confirmation

Figure 7: Workflow of feasibility upload to the user

Figure 8: Workflow of proforma invoice generation
5. Security

As user functions are accessed through public network, the application becomes vulnerable of security breach. The application user interface runs in a flash player which executes the application inside a security sandbox preventing it from malicious attacks. Also the Shockwave file(SWF) access only domain specific resources and allows access of data from that domain only. The application assets are grouped into the same security grouping. The flash player application also provides security protections for disk data and memory usage of client computer.

Encrypted password protected security mechanism is used to secure and authorized login. Standard coding procedures are used to make application robust against SQL injection and cross site scripting attacks.

6. Implementation

Front end is realized using mate flex frame work which is an event driven framework [2]. The user interaction components are designed using Macromedia Flex Markup Language (MXML) and Action Script provided by Adobe Flex. Adobe flex uses declarative model which is good for designing complex GUI with ease of reconfiguration. This application is based on RIA technology which resides on client end and enriches user experience by reducing latency in fetching results for interactive user queries.

It uses asynchronous communication and data persistence to reduce network consumption and improves overall user experience. Data integration is done via BlazeDS services which provide communication between presentation layer and business logic using remoting services.
The business logic is implemented in JAVA and various reports are generated using AlivePDF. The data access objects are implemented as POJOs for transactions with the repository. The database is implemented in Oracle 10g. Four sample screen shots [Fig 10, 11, 12] of the application are shown above.

7. Testing

This application is tested for the conformance of functional and performance requirements including security aspects. A separate test team and quality assurance (QA) team has been involved in carrying out the testing. Testing is carried out for various test cases to validate both the functional and nonfunctional requirements of the web application.

The application is tested to ensure the security of data and resources from unauthorized access. Compatibility of the application with several browsers, operating systems and screen resolutions is manually verified. The user interface is thoroughly evaluated to uncover the errors pertaining to navigation, aesthetics, syntax and semantics.

8. Conclusion and future work

The application enables faster ordering and procuring of high resolution data for Indian user which reduces turnaround time. The rich GUI enriches user experience with minimal navigation and faster response.

The application will be augmented with new high resolution foreign vendors. New administrative workflow will be realized which includes interaction among entities like purchase, stores and processing managers.

9. Acknowledgments

The authors would like to thank Director, NRSC for giving us the opportunity to realize the product. The authors are grateful to NRSC data center for providing the inputs. The authors would like to acknowledge each and every individual contributed and interfaced in realizing the workflow. The authors would also thank data processing area and quality assurance group for materializing the workflow.
10. References