DISTRIBUTED INFRASTRUCTURE ENABLING EFFECTIVE INTEGRATION OF EARTH OBSERVATION INFORMATION RESOURCES FOR COLLECTIVE SOLUTION OF ARCHIVING, SEARCHING, PROCESSING AND EO DATA ANALYZING TASKS

V. P. Savorskiy*, E. A. Lupyan², V. I. Kaevitser¹, M. T. Smirnov¹, D. M. Ermakov¹,
I. V. Balashov³, M. A. Bourtsev², V. Yu. Efremov², A. A. Proshin², V. A. Tolpin², I. A. Uvarov², E. V. Flitman²
¹Kotel’nikov Institute of Radioengineering and Electronics of Russian Academy of Sciences, Fryazino Department,
Vvedenskogo sq., 1, Fryazino, Moscow region, Russia – savor@ire.rssi.ru
²Space Research Institute of Russian Academy of Sciences, 117997, 84/32 Profsoyuznaya Str, Moscow, Russia –
evgeny@d902.iki.rssi.ru

Commission IV/5

KEY WORDS: services, integration, infrastructure, distributed, system.

ABSTRACT:

Numerous applications of Earth observation (EO) data for Earth resource exploration both for land use management purposes and for fundamental research goals require as much as possible independent EO data sets provided for a number of data consumers, i.e. for collective (shared) data usage. In addition data distribution procedures should be accompanied with sophisticated information services enabling correct interpretation and application of provided data. As a result data providers should be able to convert data in a lot of consumer-proper formats and in some cases it should require additional data processing before data shipping. Presented work proposes basic infrastructure for collective (shared) EO data usage that enables above mentioned integration of information resources contained in typical Russian EO centers. Paper reveals architecture design, description of basic infrastructure elements, and examples of real implementations.

1. INTRODUCTION

Numerous applications of Earth observation (EO) data for Earth resource exploration both for land use management purposes and for fundamental research goals require as much as possible independent EO data sets provided for a number of data consumers, i.e. for collective (shared) data usage. In addition data distribution procedures should be accompanied with sophisticated information services enabling correct interpretation and application of provided data. As a result data providers should be able to convert data in a lot of consumer-proper formats and in some cases it should require additional data processing before data shipping.

Main object of presented work is to design and implement basic infrastructure of collective (shared) EO data usage that enables integration of information resources contained in typical Russian EO centers. The principal approach in infrastructure development is based on data exchanges via wide use of unified integrated data sets (UIDS) which describe explored natural physical object or geographic zone, including observation data, products of data processing, metadata, browse images, and guides (glossaries, directories, descriptions, user manuals).

Basic element of developed infrastructure that should manage UIDS is Information Resource Integration Block (IRIB). IRIB should solve the following fundamental tasks in supporting distributed system functionality:
1) for each distributed system node that stores EO data IRIB should support all stack of services necessary to manage UIDS, i.e. data search and localization, data conversion, UIDS assembling, and data shipment,
2) IRIB should enable simultaneous access to several distributed system nodes which contain information describing the single natural physical object, so IRIB provides complex, i.e. multi-sources, UIDS,
3) IRIB should posses ability to transform data set to UIDS allows to build up unified interfaces of shared (collective) access for visualization of EO information resources as well for their on-line processing and analysis.

To integrate resources of distributed system along with IRIB the unified interface for collective (shared) usage (UICU) should be developed. UICU should enable the simultaneous receiving EO data and products as well their descriptive information metadata, browses, and guides. In addition, EO UICU allows to perform on-line reconfiguration of finalizing data processing by user request via special interface.

Paper describes architecture design supporting required service stack, shows details of basic infrastructure element (IRIB and UICU) functionality, and reveals results of real implementation of proposed approach in IKI GEOSMIS technology developed by Space Research Institute team.

2. USER REQUIREMENTS FOR DISTRIBUTED EO COLLECTIVE USAGE SYSTEM (EO CUS)

A key issue in constructing and using EO information systems is to organize the effective work of users with provided
information. To achieve this goal it is necessary to build custom interfaces that allow solving the basic problems that arise in a variety of EO systems. Naturally, every specialized monitoring system has its own special challenges, but one can outline the overall specificity, which nowadays occurs in almost all such systems. Most of modern remote monitoring systems should

1) allow to monitor large areas and quickly to provide information about monitored objects and phenomena to geographically dispersed users;
2) provide an opportunity for ongoing monitoring of controlled objects and phenomena, i.e. be designed to work with continuously updated information;
3) allow obtaining various characteristics of controlled objects and phenomena and, therefore, be focused on the use of information from various observation platform (satellites, aircrafts, in-situ etc);
4) archive data for enough long periods of observation since in many cases the objective of monitoring systems is to identify deviations from “normal” behavior of the controlled objects and phenomena;
5) be capable of working with distributed archives and databases, as the collection, analysis, processing and storage of information can take place in different geographically distributed centers;
6) due to the need of working with fairly large information flows and amounts, be mainly focused on fully automated technologies for collecting, processing, distributing and presenting the data;
7) provide users with information that has passed sufficiently deep levels of processing as well as convenient tools (interfaces for its analysis), and in many cases be capable of dynamic generation of certain products at the request of the user;
8) allow users to obtain and to analyze various kinds of spatial data;
9) effectively receive and integrate data from third-party monitoring systems as well as an easy access to various information products, including online services provided by the system.

3. EO CUS ARCHITECTURE

To work with map data EO CUS should provide at least the following functionality:

- to be able to present data on map layers;
- to own basic functions to proceed with maps;
- to be able to search catalog data and to display retrieved information;
- to be capable to can export data.

While implementing these functions in Web interfaces one should follow the next principles of their development:

- structural modularity;
- application of object data model;
- flexibility of configuration;
- easy extensibility.

As analysis reveals it is possible to split EO CUS on two functional levels which interactions are shown in Fig.1:

- **Presentation level** which should include Web and GIS interfaces and is based on usage of UICU;
- **Application level** which functionality is based on IRIB and should include:
  - Web services (including map services, metadata services, and data management services);
  - System API, enabling access to data, metadata, and various resources and interfaces.

![Fig.1. EO CUS architecture.](image)

Presentation level (aka UICU) is subsystem that supports interaction with user. This level should allow user to work with system via Web interface or via specialized applications, e.g. GIS. As typical GIS is equipped with all basic tools to work with map data its interactions with EO CUS should run on data service level. This is the reason why EO CUS should be oriented on development of unified data access interfaces, i.e. UICU. UICU architecture is presented in Fig.2.

![Fig.2. UICU architecture.](image)

Work of such interfaces should be provided by kernel consisted of two object modules:

- **Map module** which is designed in our work as smisMap software module (see Fig.2);
- **Metadata module** which is designed in our work as smisMeta (see Fig.2) software module.

**Map module** is advisable to implement as JavaScript object which should enable to display map information retrieved by data services. In addition, **module** should allow to send spatial queries and to export displaying data to print service or even to other systems.

**Metadata module** is advisable to implement as JavaScript object as well. This module should support catalog data search and retrieval of data necessary for information displaying on rendering service.
Application level (aka IRIB) is a subsystem that should enable interaction between interfaces and data themselves. This level must be consisted by Web services and system API (see Fig.3).

**Web services** should receive data access based on HTTP protocols. In order to provide application level functionality on application level of EO CUS three types of Web services must be included in IRIB:
- Map data Web service;
- Metadata Web service;
- Data management Web service.

**Map data Web services** should allow receiving map information in form of raster images. This type of services has to be based on extension of WMS standard. It should allow providing data not only for Web interfaces but as well for GIS.

**Metadata Web services** should allow receiving information on product availability for requested geographic region or dating interval as well for given additional criteria set. These services should be able to work with JSON or XML formats and send queries addressing any catalog in distributed data system. Returning information should contain both descriptive data and auxiliary data for map services.

**Data management Web services** should enable data manipulation in database via Web queries, such as adding information on contours and their attributes, their changing and deleting. Interaction with those services should be conducted in JSON format.

All three types of EO CUS Web services should be realized in frame of one common strategy i.e. should be integrated in one common service. Those common services should enable support of map data receiving (via request GetMap which should be generated in interface by smisMap module), queries on metadata receiving (via request GetMetadata which should be generated in interface by smisMeta module), requests on data addition and modification, e.g. addition of polygons and information about them. So, Web interfaces kernel should be able to provide direct interaction with metadata retrieving services and map information displaying. It is important that interactions should be possible to install with any amount of spatially distributed services.

**System API** is separate part of IRIB. It is the set of Perl modules, which should be located on servers and be able to get data access and to work with data on server side. All above mentioned services should work with data via those modules. Each module should take responsibility for definite data type or system part. It should allow flexible system configuration, error fixing, and fast software development.

### 4. IMPLEMENTATION OF BASIC ELEMENTS

#### 4.1 UICU IMPLEMENTATION

In developing the EO CUS particular attention was drawn to the issue of designing the structure of interfaces aimed to work with data sets. Taking into account needs in reasonably clear and user-friendly interfaces which should support operations with different-type quickly updated spatial data, it was necessary to develop some general approaches to allow user easily navigate interface content. To this end the basic type of Web interface was designed, namely the UICU which structure is shown in Fig. 4.

In order to support operations with different data types, UICU controls have a block structure. Each data type and typical control function matches their own module in UICU. At user level the appropriate tabs for each module and each data type are designed in UICU. This structure allows grouping various similar data and various operations on data in a convenient and clear manner. The modular structure also makes allowing UICU scalable, with no significant difficulty to extend the interface with new data types, groups, or additional operations. The structure allows configuring UICU to work with required types of data. It should also be noted that in variety of specialized systems which exploit the interfaces based on this principle, the processing control for the same type of the data can be performed in the tabs of the same type. It facilitates both the development of specialized interfaces and data exchange between different systems.

To work comfortably with multi-temporal datasets UICU realizes unified blocks which allow managing satellite data of different types as well as the results of their processing, e.g. working with:
- data of high spatial resolution displayed by quite localized scenes;
- data of coarse and moderate resolution with their characteristic units, “sessions”, covering large areas;
- series of temporal and spatial composites of various products of satellite data processing.

There’s also a special opportunity in UICU to sample different types of data layers to perform their joint analysis. Selected data layer controls are collected in a separate tab which,
describing sets servers for data access, parameters of WMS
access configuration modules contain configuration records
Configuration modules allow flexible interface setup. Data
which allow easy controlling of basic UICU settings.
Therefore the system provides various configuration modules,
optimization of creation, development and support of UICU,
Particular attention in development of EO CUS was given to
systems.
A sufficiently rich set of control to display spatial data are
supported in UICU, being almost standard for various GIS
systes.
Particular attention in development of EO CUS was given to
optimization of creation, development and support of UICU,
implementation in particular EO information systems.
Therefore the system provides various configuration modules,
which allow easy controlling of basic UICU settings.
Configuration modules allow flexible interface setup. Data
access configuration modules contain configuration records
describing sets servers for data access, parameters of WMS
queries to get the data along with additional parameters such
as data format. In metadata configuration modules parameters
of metadata receiving as well as parameters of metadata list

generation are set. smisMap configuration module allows
setting both main and additional elements of map control.
Overall configuration of the project is carried out in the main
file of the project which lists all the configuration modules.

4.2 IRIB IMPLEMENTATION

One of the main tasks of EO CUS is to unify access to different
kind of geospatial data. It stands for generalization of provided
services and standardization of data formats. The main types of
the services needed in EO CUS, like map service, map
description (metadata) service and data management service,
are integrated in IRIB. In frames of the presented work IRIB is
implemented as a Web service smiswms which implement
accesses to any kind of geospatial data in EO CUS (see Fig. 2).

The design of the service was based on OGC WMS standard
(http://www.opengeospatial.org/standards/wms), extended and
enhanced in our work in order to meet the requirements of
existing Web interfaces and peculiarity of map generation
processes. At the same time the basic format of WMS map
requests, i.e. getMap, is fully supported in our work in order to
ensure compatibility with third-party applications. Requests for
metadata and data management were also developed taking
into account the structural features of OGC standard
(http://www.opengeospatial.org/standards), targeting to the
practical applications. In particular, the metadata query can
return the results in JSON format, supports setting the size of
portions of information obtained and the work with specific
filters for various data.

Thus, the smiswms interface for data provision do not depend
on original formats of data storage and distribution. The
smiswms service, being the SW implementation of IRIB
concept, provides users access to information stored in a large
number of archives and databases in different formats. Flexible
support of these data sources and fast integration of new ones
are ensued by the features of the service internal structure. A
good example of implementation of smiswms service plug-in is
a plug-in designed to work with the distributed catalog of EO
data of high spatial resolution (Antonov et al, 2010). Access to
these data is organized in such a way that both maps and
metadata from independent catalogs of geographically
dispersed centers are aggregated “on the fly”, so it is insensible
to the users of the service.

The smiswms service has a modular structure. The modules are
grouped in fully separate specific algorithms and data access
functions from general functionality, providing the work of the
service as a whole. Thus, the service structure has standardized
groups of modules to work with particular data – plug-ins and
modules of the service core to work with plug-ins and to
support external interfaces.

Software modules, necessary for support of automatic testing of
services and data sources, are grouped separately. Key features of
IRIB implementation as the smiswms service are: separating
the processes of plug-ins and core, support of asynchronous
query processing by plug-ins, and a uniform approach to
exception handling.
The core of IRIB implementation – smiswms – performs plug-in execution and interaction with Web service through CGI interfaces, as well as functions necessary for service maintenance. Modules of each of plug-ins provide interface services of the smiswms core with data. Plug-ins are designed on the base of parent modules-classes and have standard interface functions for interaction with the core. This approach ensures easy adding and configuring of sets of plug-ins and comfortable joint development of the service functionality as a whole.

Yet another group of smiswms IRIB plug-ins forms testing plug-ins. This decision is defined, first of all, by specificity of algorithms of functional testing for each plug-in and, often, by the need of additional configuration of algorithm parameters for various projects. The testing plug-ins also share the unified interface for interaction with the core.

Much attention in development of the smiswms service as IRIB realization was paid to perform optimization. Optimization of services for data, metadata and maps provision was carried out in three main areas – optimization of algorithms and functions of data processing in every plug-in, capability of parallel execution of plug-ins, and application of means for caching queries. Nevertheless, in many respects better performance of the service is achieved by particular realizations implemented in design on specific data storage systems, which should be capable for fast data selection to generate maps and to provide metadata, including those from distributed sources. A successful approach in implementing an optimized storage system, which exploits the smiswms service plug-in to provide data access, is described in (Balashov et al., 2008).

The smiswms service implements IRIB using Perl interpreter and open source modules (see Fig. 6). This solution allowed using different operating environment without significant changes of the service.

The smiswms service implements IRIB using Perl interpreter and open source modules (see Fig. 6). This solution allowed using different operating environment without significant changes of the service.
Examples of the interfaces, implemented in the listed systems, are shown in Fig. 7. Experience of using GEOSMIS in development of these systems has shown that technology is indeed fairly universal, easy scalable and extendable for different tasks. This makes it possible to start using the technology also in other projects conducted in the Space Research Institute and Kotel’nikov Institute of Radioengineering and Electronics.

5. CONCLUSIONS

1. User requirements that determine service stack of modern EO CUS should include
   - possibilities for permanent data upgrades;
   - support of long-term data storage;
   - functionality of distributed archives and catalogues;
   - functionality implementing automatic stream-oriented data collecting, processing, distributing, and displaying technologies;
   - tools for dynamic formation of product variety by user request;
   - ability to get and analyse wide range of graphic object types;
   - ability to get and to incorporate space data from different external space data systems.

2. Proposed and motivated EO CUS architecture provides basic service stack functionality. It is worthwhile to construct and develop EO CUS as two-layer model: 1) presentation layer and application layer. Presentation layer is based on map and presentation (display) modules while application one – on data integration module and application API.

3. EO CUS is implemented as IKI GEOSMIS technology. Presentation layer is implemented in GEOSMIS map data module (smisMap) and metadata module (smisMeta). Application layer is implemented in GEOSMIS as data integration module (smiswms) and applied API.

4. GEOSMIS technology formed basis of various monitoring systems based on GEOSMIS: a) FFA ISDM, b) satellite service VEGA, c) OSM Rosrybolovstvo, d) Joint catalog at SC EOM, e) Joint system of data access of SRC of Ministry of Agriculture, f) Joint catalog of distributed satellite data archives of various centers.

ACKNOWLEDGEMENTS

The work was supported by Russian Foundation for Basic Research - grant No. 11-07-12028-ofi-m-2011.

REFERENCES


