

Data and Information Architectures for Large-Scale Distributed Data Intensive Information Systems

Larry Kerschberg, Hassan Gomaa, Daniel Menascé* and Jong Pil Yoon**

*Center for Information Systems Integration and Evolution
Department of Information and Software Systems Engineering
* Department of Computer Science*

***Present Address: Sookmyung Woman's University, Korea
George Mason University
Fairfax, Virginia, USA*

Abstract

The Earth Observing System (EOS) Data and Information System (EOSDIS) is perhaps one of the most important examples of large-scale, geographically distributed, and data intensive systems. This paper presents various facets of a data and information architecture for EOSDIS. EOS data is organized by means of an object-oriented schema, while EOS knowledge is organized through multiple domain-specific thesauri, complemented by domain knowledge and rules. The information holdings are organized into the source data archives, a data warehouse which provides an integrated view of the information holdings, and information marts which generate value-added information products for specialized user communities. Finally a federated client-server architecture is proposed to allow non-EOSDIS systems to become members of the EOSDIS community, allowing them to access EOSDIS holdings, and sharing their own data with EOSDIS.

1. Introduction

There are some important examples of information systems that are large-scale, geographically distributed, and handle very large volumes of data. One of the most important examples is the Earth Observing System (EOS) Data and Information System. EOS is a NASA program aimed at conducting a mission to study planet Earth. A series of satellites with scientific instruments aboard will be launched starting in 1997. These satellites will collect data about the atmosphere, land, and ocean. An estimated one terabyte of raw data will be sent to the Earth every day. This data will be processed to produce high-level, value-added information products for use by scientists, policy-makers, etc. The goal is to process the vast amount of data in a timely fashion and make it available to a wide user community.

George Mason University (GMU) was one of three (the others are UC—Berkeley and University of North Dakota) universities chartered to develop an independent architecture for the EOSDIS Core System (ECS). GMU assembled an *interdisciplinary* team composed of Earth scientists, computer, and information scientists. The Earth scientists of our team came from GMU's Institute for Computational Science and Informatics, the University of Delaware, the University of New Hampshire, and the Center for Ocean-Land-Atmosphere Studies (COLA) in Maryland.

The authors of this paper led the effort on the computer and information aspects of the architecture design. In particular the group specified a performance-oriented methodology to model, design and evaluate such large-scale systems in [1]. In addition, the team developed a client-server software architecture [2] for EOSDIS based on the NASA functional specifications for EOS.

This paper addresses the data and information architecture and the information services provided by EOSDIS. We focus on the issues related to the structure, function, use and management of data, knowledge and information within EOSDIS. The data and information architecture is one facet of the EOSDIS architecture, providing the infrastructure to support the processing, indexing, access, sharing and management of the vast information holdings anticipated for EOS.

2. Overview of the Earth Observing System

EOSDIS stands for Earth Observing System (EOS) Data and Information System. It is a NASA project aimed at conducting a mission to study planet Earth. It is currently the largest US-government funded science project. A series of satellites with scientific instruments aboard will be launched starting in 1997. These satellites will collect data about the atmosphere, land, and ocean. An estimated one terabyte of raw data will be sent to the

Earth every day. The entire program is anticipated to last 15 years.

Raw data sent from the satellites is first received at the White Sands complex and relayed to the Fairmont Complex in West Virginia. Then, after some initial level of calibration it is sent for archival and further processing at a collection of eight centers called Distributed Active Archive Centers (DAACs). There are currently eight DAACs: Goddard Space Flight Center (GSFC), Langley Research Center (LaRC), EROS Data Center (EDC), University of Alaska at Fairbanks (UAF), University of Colorado (CU), Jet Propulsion Laboratory, (JPL), Marshall Space Flight Center (MSFC), Oak Ridge National Laboratory (ORNL), and the Consortium for International Earth Science Information Network (CIESIN).

The raw data received by the DAACs is said to be Level 0 data. Level 0 data is used to generate Level 1 data, defined as reconstructed, unprocessed instrument data at full resolution, time-referenced, and annotated with ancillary information. Environmental variables at the same resolution and location as the Level 1 data are derived to generate Level 2 data. A set of variables mapped on uniform space-time grid scales, with some consistency and completeness, are called Level 3 data. Finally, the model output or results from analyses of lower level data is known as Level 4 data. Not all data products needed to generate high-level products will be located at a single DAAC; in fact, there are complex product interdependencies and this means that data products will be sent from DAAC to DAAC for processing.

About 500 NASA-selected scientists will determine which standard data products are to be generated by the EOS Core Systems (ECS). The facilities where these scientists are located are called SCFs (Science Computing Facilities). The remaining sections of this paper concentrate on the EOSDIS Core System (ECS), and in particular on its Science Data Processing Segment (SDPS).

The organization and management of data, knowledge and information will be of utmost importance to the success of EOSDIS. We have adopted the convention of the push-pull paradigm associated with information processing. The *push* aspects refer to capturing the raw radiances and converting them to products, while the *pull* aspects refer to the EOS user community accessing and manipulating those products.

3. EOSDIS data and information architecture

The data and information architecture presented in this paper is influenced by research and development work by

the authors on several major efforts in: 1) the Intelligent Integration of Information [3, 4, 5, 6, 7], 2) the Active Data/Knowledge Dictionary [8], and 3) the Evolutionary Domain Life Cycle Model [9, 10] and the associated prototype software, the Knowledge-Based Software Engineering Environment (KBSEE) [11, 12], which was used to specify the Domain Model of ECS [13].

3.1 Data architecture

The data architecture presented in this section is based on the object-oriented requirements analysis conducted for the specification of the ECS Domain Model [13]. The object-oriented specification diagram notation is based on the technique proposed by Rumbaugh [14]. Figure 1 shows the various symbols used in the Object Modeling Technique (OMT).

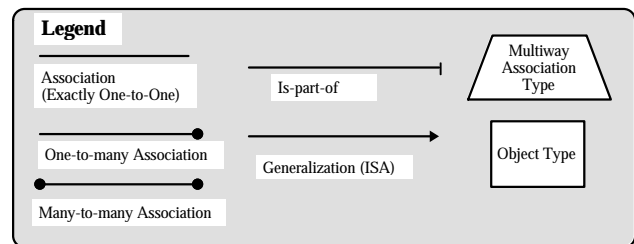


Figure 1: Legend for OMT schema diagrams

EOSDIS real-world concepts. Figure 2 represents concepts (object types or classes) that correspond to the “real-world” as well as those produced by ECS. Note that a rectangle represents an Object Type or Class, and a polygon represents a multi-way association type. Binary association types (exactly one-to-one, one-to-many, and many-to-many) are also shown; two special associations are the *Is-part-of* and the *generalization* (ISA) associations.

The classes such as Mission, Satellite, Instrument, Principal Investigator and EOSDIS Facility represent real-world concepts. An *instance* (or object) of a certain type is a member of that type. For example, the MODIS instrument is an instance of the object type Instrument.

The *Is-part-of* association indicates that an object type may be composed of constituent object types. In Figure 2, Instrument Data are composed of Instrument Engineering Data and Raw Radiance.

In Figure 2, the EOSDIS Facility type has two subtypes: ECS Facility and Non-ECS Facility. The ECS Facility has two subtypes: Distributed Active Archive Center (DAAC) and Data Warehouse (to be defined shortly).

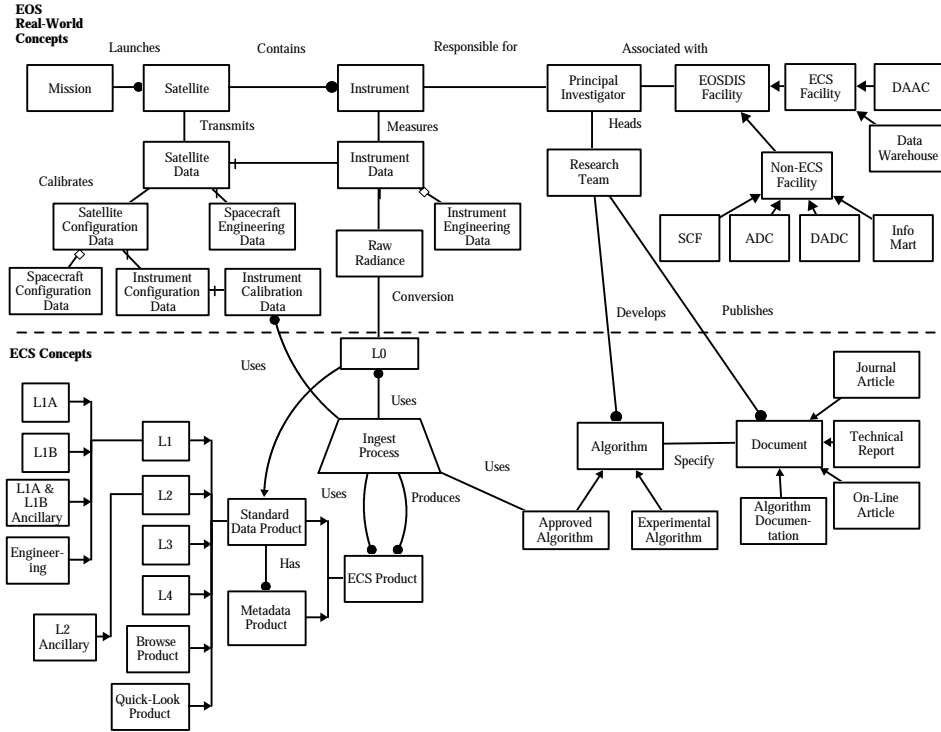


Figure 2: EOSDIS data schema

EOS core system concepts. The EOS Core System processes Level 0 products. These, in turn, are processed further using algorithms provided by research teams headed by principal investigators. Note that an algorithm may be of two types, approved or experimental. The distinction is that the approved algorithms are used to produce ECS Products, while the experimental algorithms are undergoing testing for eventual use in the Ingest Process.

The Ingest Process uses Level 0 products to produce higher-level products, and these in turn are used to produce instances of both Standard Data Product and Metadata Product. The Standard Data Product Object Type is specialized into L0-L4, Browse Product, and Quick-Look Product. The L1 product itself is further specialized to L1A, L1B, L1A & L1B Ancillary, and Engineering; L2 is specialized to L2 Ancillary.

Additional ECS-specific concepts are associated with Documents published by Research Teams; these include Algorithm Documentation, Journal Articles, Technical Reports, and On-Line Articles.

EOSDIS knowledge schema. The meta-level layer of the data architecture is shown in Figure 3. It represents the knowledge and metadata needed to support the information view of EOSDIS. The Meta-Level Layer contains distinct types of metadata and knowledge. The Hughes Data Pyramid spans L0 through L4 Standard Products, as well as other metadata typically associated

with Earth science data sets.

We propose a Global Thesaurus of concepts associated with EOS in general, and ECS in particular, so that casual users as well as sophisticated users may access the information holdings of EOS in *flexible* and *multi-faceted* ways.

Lastly, we propose to enhance the Meta-Level Layer with *knowledge sources* — both domain-specific and domain-independent — that can be used by EOSDIS in a variety of ways: 1) to cooperate with users in formulating queries, 2) to maintain subscription lists for ECS products, 3) to optimize the performance of one or multiple queries, and 4) to support

generalized *active* object views that are automatically updated or materialized, based on conditions monitored in the respective ECS databases. These will be discussed in the sections which follow.

Global thesaurus. The Global Thesaurus serves the same purpose that a thesaurus serves in a library, that of assisting users in identifying similar, broader and narrower terms related to a particular search term. This enables the user to obtain broader terms, narrower terms, similar terms, etc., thereby increasing the likelihood of obtaining the desired information from a repository of information holdings.

The major distinction between a library thesaurus, and the proposed Global Thesaurus is that active rules may be associated with object types as well as their attributes and functions. This is represented in Figure 3 by the Uses associations between the EOS Global Thesaurus and EOS Knowledge Base object types and between the EOS Global Thesaurus and the Hughes Data Pyramid object types. The Global Thesaurus is comprised of specialized thesaurii as depicted in Figure 3.

EOS knowledge base. The EOS Knowledge Base consists of several different types of knowledge: Domain Knowledge, Constraints, and Active Rules and Heuristics. The EOS Knowledge Base is composed of different types of knowledge: 1) domain knowledge regarding missions, instruments, satellites, etc., 2) constraints such as data

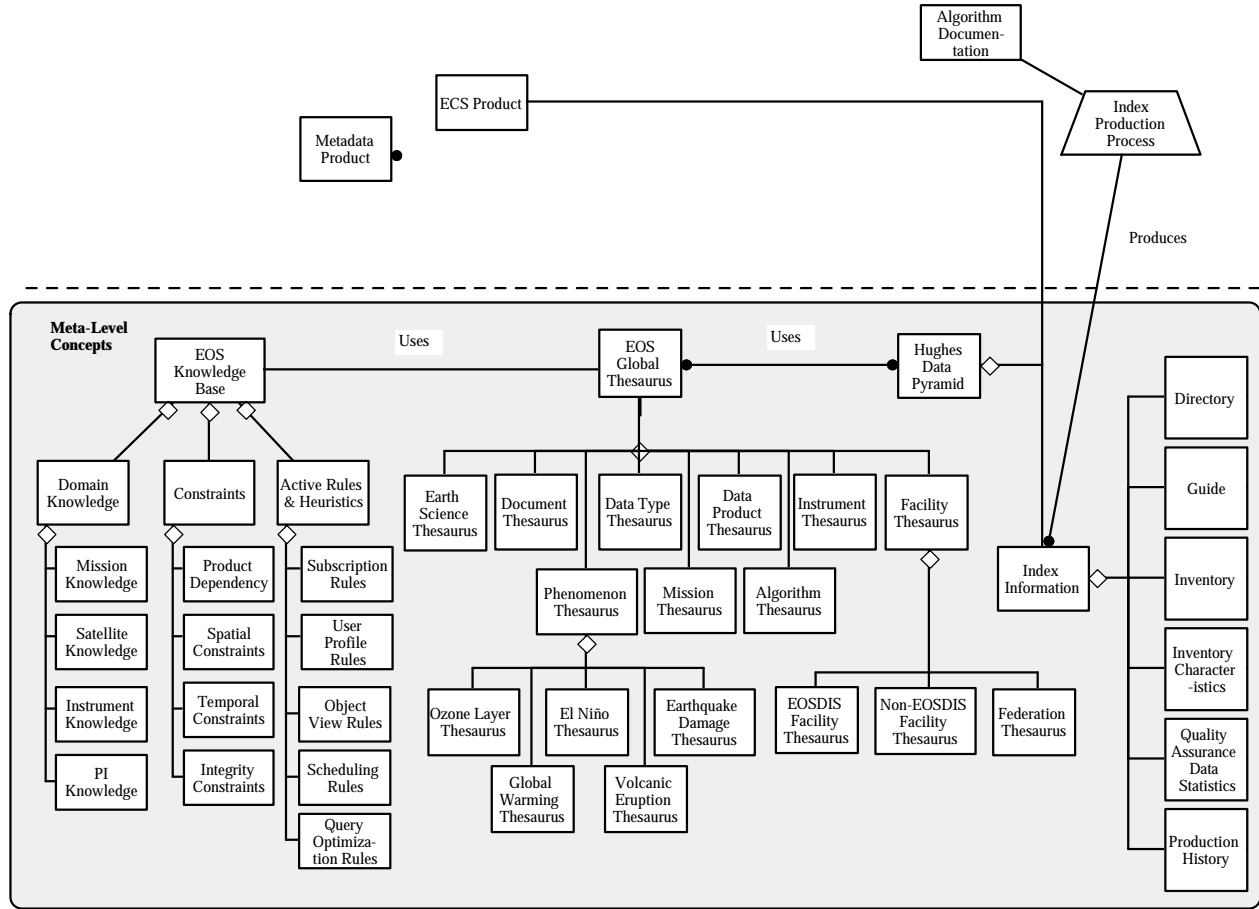


Figure 3: EOSDIS Knowledge Sources

dependencies associated with products derivation, spatial and temporal constraints, as well as integrity constraints, and 3) active rules and heuristics. This knowledge can be used in conjunction with the Global Thesaurus as follows: 1) to enhance the ability of users to obtain domain-specific knowledge about missions, satellites and their instruments, and PI information, 2) to allow users to examine EOS constraint knowledge in the form of product dependency information, temporal and spatial constraints on missions, orbits, satellites and instruments, and 3) to support modular knowledge sources consisting of rules and heuristics in support of user subscriptions and profiles, the materialization of complex object-views, and for scheduling and query optimization.

3.2 Multilayer information architecture

The information architecture proposed is multi-layered and reflects the data and information needs of the EOSDIS user community. This includes: 1) those involved in the production of ECS Standard Products, 2) principal investigators at SCFs are interested in value-added derived products, 3) users with specialized

interests, such as, students and instructors in grades K-12, and 4) casual users who wish to learn more about EOSDIS. In order to serve these EOSDIS constituencies and balance the demands on ECS resources, we use the architectural concepts of the Data Warehouse and the Info Mart.

Data warehouse. The goal of creating a *data warehouse* is to *extract* data from multiple production databases or information systems, for example ECS Data Products archives at DAACs and other closely associated facilities such as SCFs, and to organize that data according to a *common schema*, so that the entire user community may access that data for decision-making and for the production of value-added products.

The EOSDIS data warehouse would support “popular” ECS Data Products and contributed products from other EOSFed members (EOSFed is discussed in the next section). These would include the most sought after and timely products. Other products would reside in archives as shown in the archive layer of Figure 4. Note that those products not available in the data warehouse could still be obtained through on-demand processing requests.

The processes of creating, populating, and managing the data warehouse would be handled by the information management system and the federation services. The data from federation members would adhere to the standards set forth by EOSFed for the data warehouse, and appropriate data translations and schema integration would be performed to obtain the common object-oriented schema for the warehouse. The schema would be based on the one shown in Figure 2. A Data Warehouse would probably reside at a DAAC.

Info mart. The info mart differs from the data warehouse in that it specializes in providing knowledge-rich products to restricted user constituencies. In the context of ECS, each info mart would serve a specific purpose and would have a well-defined user community. Thus, K-12 students and instructors would access their info mart to obtain the latest weather maps, images to be used in conjunction with study guides, etc. Similarly, policy makers could access reports, images and prepared multi-media presentations associated with Global Warming and the Ozone Layer, while Earth scientists interested in El Niño phenomenon could subscribe to the latest products of their info mart.

4. EOSDIS federated architecture

A *cornerstone* of the GMU architecture is the concept that EOSDIS is actually a *federation of heterogeneous, autonomous* information systems. They may be *heterogeneous* in that each system may have different hardware, possibly different database management systems and other software systems, and certainly different kinds of data products.

Further, each system will want some degree of *autonomy* over its operating schedule and its level of participation within the federation. Usually, an autonomous system will want to support local users first and federation clients second; in order to allow local autonomy while still ensuring that EOSDIS service levels be met, we introduce the concept of *degree of coupling*, to be defined in the next section. The proposed solution for managing EOSDIS system heterogeneity and autonomy is to create a *federation* of client/server systems.

4.1 Definition of EOSFed

The **Earth Observing System Federation** is a client-server federation of EOSDIS and related systems which:

1. subscribes to the common membership rules of the federation,
2. provides services to other members in terms of access to, or on-demand processing of, data products,
3. allows federation users to browse the entire federation holdings by means of the Global Thesaurus,

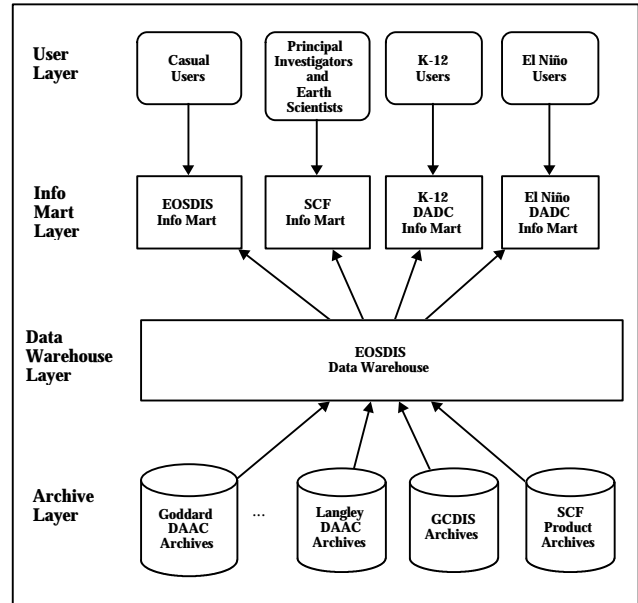


Figure 4: EOSDIS multi-layer information architecture

4. supports user query formulation and reformulation to access data products from multiple Info Marts, the Data Warehouse, databases and product archives, and
5. supports client-server interactions among EOSFed members.

The Federation Manager (FM) depicted in Figure 5 is responsible for providing services to the entire EOSFed. Note that these services may migrate and also reside at client or server sites, should performance considerations warrant such migration.

Second, each EOSFed member has a Federation Interface Manager (FIM) as shown in Figure 5, whose role is to shield the member information system from having to know about the rules and regulations of EOSFed. The FIM is the software subsystem that manages each member information system's interface to the federation, and serves as its *agent* or mediator as proposed by Wiederhold [15] for interactions with other EOSFed members. Thus, the member information system can retain its *autonomy* and *responsibility* for its information holdings while still participating in EOSFed.

Further, the FIM handles the various client-server sub-federation commitments of the local system, and contains sufficient knowledge to represent the local system's holdings in the *common object model* shared by all members of EOSFed. In addition, the FIM may have all or a portion of the Global Thesaurus and its associated software so that local users may access the EOSFed resources.

We now present the various EOSFed services as supported by the Federation Manager and the Federation Interface Managers.

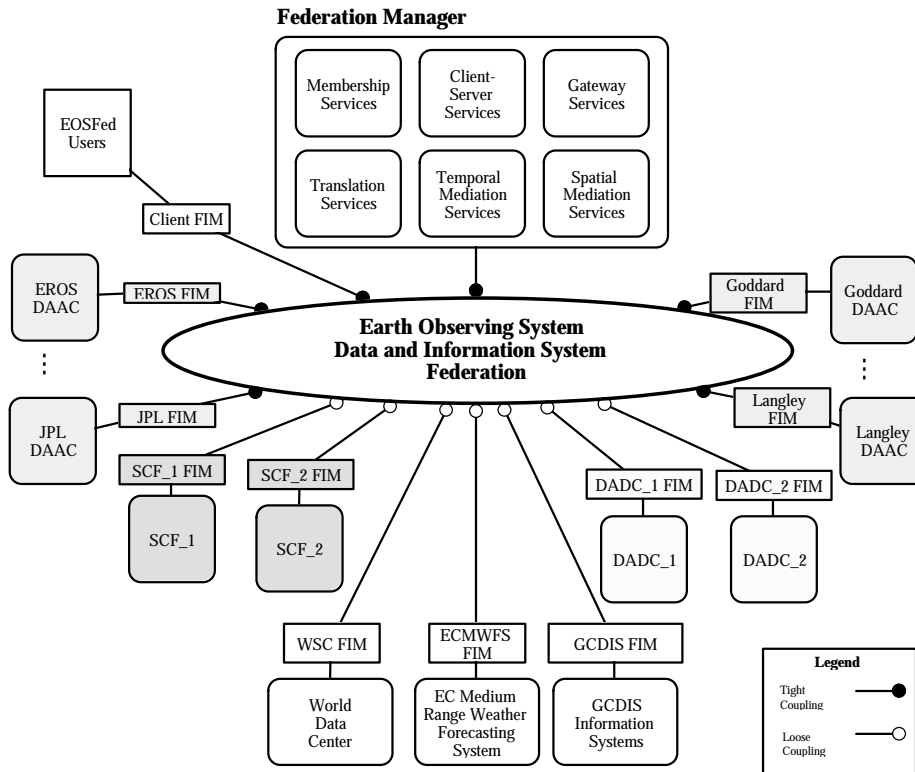


Figure 5: The EOSFed Architecture

4.2 Membership services

EOSFed and its sub-federations will provide services to its members. Members, in turn, have responsibilities to the federation as well. These include, but are not limited to the following:

- Rules and regulations for creating a federation, and eventually terminating a federation.
- Rules to become a member of a specific federation.
- Rules to leave a specific federation.
- Mechanisms to change the bylaws of a federation.

We now examine each of these topics in more detail. First, a new federation may be created when the community decides that a need exists for the formation of the federation. There are several steps involved in creating the federation. First, an application area is identified, e.g., ECS Data Product Processing. The DAACs and SCFs might be members of such a federation, and the dependency graphs associated with certain data products might specify the particular client-server arrangements among them. The next step is to specify the information that is going to be exchanged and processed by the members including transaction types and query types, protocols, message content, and the message format.

Members may subscribe to different levels of services and different types of coupling for each sub-federation to which they belong. EOSFed Services are available to all members.

4.3 Client-server services

Each Client-FIM consists of three subsystems, a Client Router, Client-to-Federal Translation Services, and Federal-to-Client Translation Services. The translation services map local data objects to the EOSFed representation and vice-versa. The Client-FIM 1) accepts local transactions (including query requests) from local clients in the format used by the local information system, and 2) the router determines whether the transaction is for a local server or a remote server of the federation. If the destination is a valid server, the request is passed to the Client-to-Federal

Translation Services so the request may be translated to standard federal transaction format prior to routing it to the appropriate server.

The Server-FIM consists of Federal-to-Server Translation Services, Server-to-Federation Translation Services, and the Server Router. When a transaction arrives at a destination, the Federal-to-Server Translation Services translates the federal transaction into the local transaction format of the server.

After translation the transaction is sent to the server router. The router receives all transactions, both locally and remotely generated, and logs them. Transactions are then queued for processing. Once the server has processed the transaction, the server router sends the response, if necessary, to the Server-to-Federal Translation Services, for server to EOSFed translation, and then routes the response to the appropriate Client-FIM, where the response is translated into the local format.

4.4 Translation services

EOSFed Translation Services effect the translation from the local information system to the EOSFed data/knowledge structures. It is important to stress that both the *syntax* and *semantics* of objects be translated for

effective communication among federation systems.

Data language translation services encompass both syntax and semantics. Syntactic translations are primarily data manipulation language translations, but also include translations for data definition languages used when adding new constituents to the federation. Semantic translation services include those required to resolve structural heterogeneities (e.g., type mismatch, format differences, unit/measurement inconsistencies, and differences in constituent term granularities) and the operational (behavioral) characteristics of the constituent objects as discussed by Weishar [15, 16, 17, 18].

The translations required to handle the mismatch are provided directly by specialized EOSFed *knowledge modules* (KM) contained within each FIM and specialized to the specific needs of that federation constituent, or indirectly by providing access protocols to specialized federation translation and mediation knowledge sources maintained by the Federation Manager. These would handle the translation of subqueries from EOSFed to the local query language, as well as the data format translations between the local systems "objects" and the corresponding EOSFed objects. Note that these knowledge modules may also be available at the FIMs and can be dynamically updated as required.

Data/knowledge translation services are also required so that various tools and applications may access data and knowledge in the representations most suitable to their needs. Thus, EOSFed Translation Services must support a variety of data/knowledge representations, including: rule-based, object-based, semantic net-based, and relational database models.

4.5 Temporal mediation services

EOSFed databases will have data sets and objects with time-varying attribute values together with time stamps that indicate the periods of validity for these values. Time-stamped data often exhibit more complicated syntax and semantics than conventional non-temporal data. This leads to at least two problems: (1) users have to understand possibly different logical and physical structures of the underlying data in these databases, and (2) user queries must be in terms of the time units (e.g., hour, day, month) on which the time stamps are based. These two problems may prevent a user from using the databases effectively. In order to solve these two problems, EOSFed will have a temporal mediation service provided by a temporal mediator [19].

4.6 Spatial Mediation Services

The standard for digital geospatial data [20] provides guidelines for both temporal and spatial information for such data sets. The *spatial mediator*, similar to the

temporal mediator described above would handle the translations among various spatial coordinate systems, and provide an EOSFed representation of the spatial data. There could be associated with the spatial mediator a collection of models to process the data and to provide the integration of information from multiple sources, such as *in situ* measurements, satellite radiances, buoy measurements, etc.

4.7 Example of Temporal and Spatial Mediation

As an example of both temporal and spatial mediation services, we describe the problem of *data assimilation* which is one of the key EOSDIS applications. Its goal is to acquire Earth science data from several sources such as EOS instruments, non-NASA satellites, and from several other sources including ground sources, and provide this data in a correct and consistent form as input to comprehensive Earth system models [21, 22]. There are several challenging problems to be solved in data assimilation for EOSDIS including the vast quantities of data, the different data formats, inconsistencies in the data, including missing and incorrect data.

EOSFed would provide a means for interfacing to the different data sources, with their different formats, access patterns, units, etc. as input, and providing a standardized EOSFed output (units, formats, spatial coordinates, time units), as required for a given experiment. The same data could, if necessary, be prepared differently for another experiment. Each data source, such as EOS instrument data, NOAA satellite data, DSMP satellite data, European Space Agency satellite data, cloud motion winds, surface data, aircraft, ship and rocketsonde reports [22], would be supported by its own Federation Interface Manager, specialized to address that data source's data structures, and carry out all data preprocessing to map the data to EOSFed format, resolving any inconsistencies and compensating for any missing data in the process. The data would then be analyzed and piped into the scientific model. In the case of the assimilation system for the atmospheric general circulation model (AGCM), the federation format would consist of gridded spatial and temporal meteorological data.

5. Conclusions

This paper has presented the data and information architecture for a large-scale data-intensive system, the Earth Observing Data and Information System. The architecture combines data, knowledge and information oriented concepts which are organized at various levels. Further, EOSDIS will support a federation of information sources and systems, and our paper addresses the services supported by such a system. These architectural concepts are applicable to other large-scale data-intensive systems.

6. Acknowledgments

The authors would like to acknowledge the many useful discussions with the independent architecture study group for EOSDIS ECS at GMU led by Menas Kafatos. In particular, the authors would like to thank Jim Churgin, Ferris Webster, Berrien Moore III, and Jim Kinter, for explaining to them the different aspects of Earth science and user scientific requirements for EOSDIS. Special thanks go to Frank Carr who provided important comments on the EOSFed architecture.

7. References

- [1] Menascé, D., H. Gomaa, and L. Kerschberg, "A Performance-Oriented Design Methodology for Large-Scale Distributed Data Intensive Information Systems," *Proc. First IEEE International Conference on Engineering of Complex Computer Systems*, November 7-10, 1995, Fort Lauderdale, FL (Outstanding Paper Award in Systems Engineering Track).
- [2] Gomaa, H., L. Kerschberg, and D. Menascé, "A Software Architecture Design Method for Large-Scale Distributed Data Intensive Information Systems, ISSE Technical Report, 1995.
- [3] Kerschberg, L. and Yoon, J.P., "Semantic Query Reformulation in Object-Oriented Databases," *Proc. Workshop on Combining Declarative and Object-Oriented Databases*, Washington, DC, May, 1993.
- [4] Seligman, L. and Kerschberg, L., "Approximate Knowledge Base/Database Consistency: An Active Database Approach," *Proc. First International Conference on Information and Knowledge Management*, November, 1992.
- [5] Seligman, L. and Kerschberg, L., "An Active Database Approach to Consistency Management in Heterogeneous Data- and Knowledge-based Systems," *International Journal of Cooperative and Intelligent Systems*, Vol. 2, No. 2, October, 1993.
- [6] Yoon, J.P. and Kerschberg, L., "A Framework for Constraint Management in Object-Oriented Databases," *Proc. First International Conference on Information and Knowledge Management*, November, 1992.
- [7] Yoon, J.P. and Kerschberg, L., "A Framework for Knowledge Discovery and Evolution in Databases," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 6, December 1993.
- [8] Dove, J., L. Kerschberg, P. Berra, C. Bosch, D. Weishar, A. Waisanen, J.P. Yoon, "Active Data/Knowledge Dictionary," Final Technical Report, RL-TR-91-231, Rome Laboratory, Griffiss Air Force Base, NY, 1991.
- [9] Gomaa, H., L. Kerschberg, V. Sugumaran, "A Knowledge-Based Approach for Generating Target System Specifications from a Domain Model," *Proc. IFIP World Computer Congress*, Madrid, Spain, September 1992.
- [10] Gomaa, H., L. Kerschberg, and V. Sugumaran, "Knowledge-Based Approach to Domain Modeling: Application to NASA's Payload Operations Control Centers," *Telematics and Informatics*, Vol. 9, Nos. 3/4, pp. 281-296, 1992.
- [11] Bosch C, H. Gomaa, L. Kerschberg, "Design and Construction of a Software Engineering Environment: Experiences with Eiffel," *IEEE Readings in Object-Oriented Systems and Applications*, IEEE Computer Society Press, 1994.
- [12] Gomaa, H., L. Kerschberg, and V. Sugumaran, C. Bosch, I. Tavakoli, "A Prototype Domain Modeling Environment for Reusable Software Architectures," to appear in *Proceedings of Third International Conference on Software Reuse: Advances in Software Reusability*, Rio de Janeiro, Brazil, November 1-4, 1994.
- [14] Rumbaugh, James, et al, *Object-Oriented Modeling and Design*, Prentice Hall, Englewood Cliffs, NJ, 1991.
- [15] Weishar, D., and L. Kerschberg, "An Intelligent Interface for Query Specification to Heterogeneous Databases," NSF Workshop on Heterogeneous Database Systems, Northwestern University, Evanston IL, December, 1989.
- [15] Wiederhold, G., "The Roles of Artificial Intelligence in Information Systems," *Journal of Intelligent Information Systems*, Vol. 1, No. 1, Kluwer Academic Publishers (L. Kerschberg, Editor-in-Chief), August 1992, pp. 35-55.
- [16] Weishar, D., and L. Kerschberg, "An Intelligent Heterogeneous Autonomous Database Architecture for Semantic Heterogeneity Support," *IEEE International Workshop on Interoperability in Multidatabase Systems*, Kyoto, Japan, April, 1991, pp. 152-155.
- [17] Weishar, D., and L. Kerschberg, "Data/Knowledge Packets as a means of Supporting Semantic Heterogeneity in Multidatabase Systems," Special Issue: Semantic Issues In Multidatabase Systems, *SIGMOD Record*, Vol. 20, No. 4, December, 1991.
- [18] Weishar, D., "A Knowledge Based Architecture for Query Formulation and Processing in Federated Heterogeneous Databases," Doctoral Dissertation, George Mason University, Spring 1994.
- [19] Jajodia, S. and Wang, X.S. , "Temporal Mediators: Supporting Uniform Access to Heterogeneous Temporal Databases," Workshop on Interoperability of Database Systems and Database Applications, Friburg, Switzerland, Oct 13-14, 1993.
- [20] "Content Standards for Digital Geospatial Metadata," Federal Geographic Data Committee, June 8, 1994.
- [21] Rood, R. B., and Stobie, J.G., "Data Assimilation and EOSDIS," NASA Internal Report, November 1993.
- [22] Schubert, S.D., R. B. Rood, and J. Pfaendtner, "An Assimilated Dataset for Earth Science Applications," *Bulletin of the American Meteorological Society*, Vol 74, No. 12, December 1993.