Summary and Analysis of the GMU Independent Architecture Study

The George Mason University (GMU) team was composed of an interdisciplinary group of earth scientists and computer scientists. The team identifies a number of sources of stated earth scientist's needs, including some of the key sources developed by the ECS team in its architectural redesign effort. The GMU team adds their own observations, and develops a broad architectural framework capable of supporting these needs. Like the current ECS approach, much of the focus of the GMU team was aimed at developing a flexible architecture for ECS that can accommodate a growing level of demand from both science and non-science users.

The fundamental solution recommended in this report is to create facilities referred to as Domain Application Data/Distribution Centers (DADC's) which can service the unique needs of specialized user communities. While the establishment and funding of such facilities is a programmatic issue, the GMU team addresses the ability to support such a concept from an architectural perspective. This is quite similar to the ECS team's approach at developing a "policy neutral" architecture that allows data provider policies to be established independently of the core architecture development.

As a result, the GMU team has arrived at an architectural framework that is conceptually similar to the current ECS design, with the built-in flexibility to meet the requirements of ECS under the current programmatic framework, but which also will be able to evolve to support the inclusion of a broader group of researchers and users whose needs are not specifically addressed by the current scope of EOSDIS. Ultimately, the inclusion of these communities will also return value back to the science users of EOSDIS by providing them with more ready access to data products that are generated outside of the normal "production" system.

The earth scientists on the GMU team explored several scenarios in each of three earth science research areas: terrestrial, oceanographic, and meteorological. Each of these scenarios describes a variety of aspects of user interaction with the system that have important implications for ECS architecture designers. Not only do these scenarios describe various types of query sequences that users will make individually, they also describe various forms of collaboration and data sharing between users that must be supported by the architecture. Throughout these scenarios, one theme seems to dominate: most serious earth-science research will require a combination of data from sources both inside and outside of the supported EOSDIS infrastructure. It is therefore vital to these scientists that the ECS architecture provide them with an effective means to access all relevant data, regardless of the source. Another item worth noting is an anecdotal description in Part 2, Chapter 5 of a case in which an algorithmic flaw created erroneous data that propagated errors into many high-level data products. The discussion describes a number of issues that must be considered in regard to identifying, tracking down, and rectifying such inevitable errors.

The computer scientists on the GMU team developed a federated client-server architecture that is intended to address the key science drivers identified by the earth scientists. An important aspect of the client-server architecture is that it is composed of standardized "ECS nodes" which may be individually configured to meet varying levels of processing and archival requirements.

One of the more unique aspects of the GMU team's information architecture is the emphasis on incorporating and organizing knowledge in the system in addition to data. This provides for the organization of "info-marts" for value-added, knowledge-rich specialized products. Issues of knowledge-based information access are addressed through the notion of a Global Thesaurus. Tying it all together is their EOSFed concept which describes the structure and functions needed to allow autonomous and heterogeneous systems to become EOSDIS members. These concepts directly parallel the concepts of value-added providers, domain specific data servers, and smart data dictionary present in the ECS architecture.

Other important aspects of the architecture development were the modeling and analysis techniques used by the GMU team. Custom-modified COTS domain analysis tools were applied to characterize a family of systems that are independent of a specific mission, instrument suite, or hardware configuration. Using this domain model in combination with a functional description of the architecture, a performance modeling methodology was used to illustrate how a proposed architecture can be analyzed. Using these analysis techniques, it is possible to determine the impact that various alternative system configurations might have on response time. Another important form of analysis provided in this report is a set of event-sequence diagrams of various user scenarios, showing how the proposed system responds to user requests. GMU recommends that the modeling approaches and expertise of the GMU team could be used to augment ECS modeling efforts.

To complement the architecture development, studies are provided on ATM networks, data storage systems, and high performance database management systems. These studies outline the current state of the art, and make specific recommendations for the ECS architecture. These recommendations are, in large part, consistent with the ECS team's stated technology adoption plans as put forth at the System Design Review (SDR).

Finally, the report also addresses issues of management of large science projects and how to effectively integrate EOSDIS into the broader context of projects such as GCDIS. The report stresses the need to make data and services available to the widest possible user population in order to achieve maximal success for the overall program. This was a common theme from the teams, and suggests that the ECS team should be quite agressive in defining its value-added provider (VAP) interfaces.

Below are some of the key recommendations for the ECS program based on the GMU report:

Information Management

- **Re-architect DAAC roles and add DADC's to provide specialized support consumer-oriented services** :
  Categorization: PRG

  A primary element of the GMU approach is the notion that the architecture must be able to support Domain Application Data/Distribution Centers (DADC's). The role of the DADC is to provide services and high-level EOS data products to specialized communities of users. The DADC's will serve both to offload the DAAC's from intensive pull demands from thousands of non-science users as well as to improve the access to non-EOS data for the science users. (Part 5, Ch. 1)

  This concept is consistent with the ECS architecture concept of "Value added providers". The ECS architecture envisions SCFs and other data centers as being able to fulfill this role by taking primary feeds from the DAACs, re-packaging data and services, then delivering those to it's own community through a variety of interfaces. The movement of production responsibility for the "standard products" between DAACs, SCFs, and other providers, is a programmatic / policy issue.

- **Use of Global Thesaurus concept** :
Categorization: ECS, R&D, EVO(S/L)

The GMU architecture introduces the concept of a Global Thesaurus whose purpose is to provide a mapping from a user's term into both broader and narrower terms, thereby increasing the likelihood of obtaining the desired information from a repository of information holdings. Unlike a standard thesaurus, the Global Thesaurus makes use of active rules to aid in the translation of terms in different domains.
(Part 3, Ch. 4, pg. 7)

This concept is partially accommodated by the ECS architecture's Data Dictionary component. However, the GMU team proposes an innovative approach to thesaurus configuration and management at both the local and global levels of the system. Their recommendations should be investigated further as part of the ECS design. The feasibility of establishing a truely global thesaurus remains to be established. The successful implementation of this concept hinges on a series of technical, behavioral and managerial issues.

- **enhancement of data browsing and query processing** :
Categorization: ECS, EVO (S/L)

The primary role of the Global Thesaurus is to improve data browsing and query processing functions. Queries made by the user will often not match items in the database, simply because the terms used in the query are not the same as those associated with the data. The Global Thesaurus will make this access more effective. Some of the issues involved in providing an intelligent thesaurus require complex semantic mappings. Such issues will be addressed in an evolutionary fashion within ECS.

- **scalability of Global Thesaurus to a large distributed system** :
Categorization: R&D

While GMU has initiated work on a Global Thesaurus, further research is needed to demonstrate that this concept can be scaled to a large distributed system. A number of the research issues that must be addressed are similar to those of large scale, distributed information systems, with added complexities in knowledge representation, concept definition and refinement, etc.

- **creation and maintenance of the Global Thesaurus** :
Categorization: R&D, PRG

Further research is also needed to show how a Global Thesaurus might be created and maintained by a large scientific community. The more people that get involved in this process, the more difficult it becomes. From a reseach perspective, there are three fundamental problems. Tools are needed to semi-automatically capture domain concepts and the knowledge that links them together. Also, tools are needed to maintain consistency of the knowledge. These are difficult problems even without distributing the knowledge base, so implementing a global solution in a distributed environment will be particularly challenging. Additionally, there will be programmatic issues involved in resolving semantic terms and mappings proposed by different segments of the ECS user community.

- **Use a federated architecture to allow integration of heterogeneous components (multi-database support for the EOSFed concept)** :
Categorization: ECS, EVO (S), R&D

The GMU approach advocates a federation of heterogeneous and autonomous information systems. Therefore, they have concentrated on issues related to supporting varying degrees of coupling to the federation, management of the federation, and provisions of mediation services that allow translation of information between a local information system and the rest of the federation. (Part 3, Ch. 4, pg. 31)

This recommendation is consistent with current ECS plans to support heterogeneous, autonomous providers. Federation approaches and issues raised by the GMU team are incorporated in the short and long term design plans of the ECS team.

- **Pre-query cost estimation using meta-data :**
Categorization: ECS, EVO (S/L)

If the size of a response to any query exceeds a user-determined threshold, the GMU architecture is designed to provide a report to the user indicating the time that would be required to deliver this information as well as options for different delivery alternatives with their associated costs. (Part 3, Ch. 1, pg. 5)

This recommendation is consistent with current ECS plans to support provider service descriptions (including costs), as well as service-based accounting and mediation services. The GMU team's suggestions should be further incorporated into the ECS team's design and implementation efforts.

- **Provide interoperability with data sources outside of EOS** :
Categorization: ECS, EVO (S/L), PRG

GMU scenarios emphasize the need for scientists to be able to easily access data sources outside EOS. The DADC concept recommended by GMU provides a means to support this outside access.

A significant degree of external system interoperability is incorporated in the ECS architecture, primarily as a means of accessing ancillary and correlative data for purposes of product generation. ECS is required to be interoperable with Ancillary Data Centers involved in production processing. The GMU team recommends a broader definition of interoperability, one that would enable users to make transparent queries across multiple data and information systems(ECS plus other systems). In addition to requiring technical work to support this in an evolvable fashion, this issue requires some programmatic direction from ESDIS.

- **Generic ECS node architecture that can be specialized to support different user requirements** :
Categorization: ECS, EVO (S)

The building block of the GMU architecture is called an ECS node, which is a collection of servers of different types. ECS nodes can be configured to serve as DAAC's, DADC's, ADC's, or even SCF's. (Part 3, Ch. 2, pg. 6)

The GMU team specifically decomposes the ECS node into distinct server "types", which it appears to make available through site interfaces. The ECS architecture supports a similar implementation, however it strives to unify the interface into those servers through a common gateway interface called the Data Server. This provides the "federation" mechanism alluded to by the GMU team, and allows individual sites to decide which servers and services it wishes to make available.

- **Archival of data in a variety of user-oriented forms** :
Categorization: EVO (S)

GMU proposes the Info Mart and Data Warehouse concepts as a means of extracting and organizing data according to a common schema so that specific user communities will be able to access the data for decision making, and to encourage the production of domain-specific value-added products. (Part 3, Ch. 4, Pg. 12)

In fact, this concept is also consistent with ECS architecture and evolvability approaches. Value added providers will receive primary feeds from the ECS DAACs, and will then organize the data according to the needs of their own communities.

- **Layering of the architecture to separate push and pull requirements** :
Categorization: EVO (S), PRG

One of the key points behind the use of DADC facilities is that the DADC's can service the specialized pull requirements of specific user communities, thereby allowing the DAAC's to focus on their role of handling the push requirements of data processing and archiving.

Again, such separation is achieved in the ECS architecture through the Value Added Provider (VAP) network. The development and evolution of VAPs over time will be driven by individual user community data needs. The provision of Government funds for the creation of VAPs is a programmatic issue.

- **Methodology for performance analysis of large distributed client-server architectures (do performance modeling scientifically)** :
Categorization: ECS, EVO (S)

GMU has applied a mixed queuing network modeling approach for analyzing performance of prospective architectures. This technique permits scientific studies of different architectural alternatives, and provides quantitative estimates of system response time and sensitivity to changes in workload.
(Part 3, Ch 5)

The ECS team should augment it's performance modeling activities to include the sorts of studies advocated and demonstrated by the GMU team, especially for the "pull" side of the system.

- **Query formulation and optimization that uses metadata and thesaurus information** :
Categorization: ECS, EVO (L), R&D

Provide cooperative query formulation services in which metadata and knowledge are used to assist users in formulating and re-formulating requests. Allow for both generalization and specialization in both the temporal and spatial dimensions.
(Part 3, Ch. 4, pg 19)
  - **construction and use of query optimization rules (for extended types)**
  - **cooperative query formulation**
  - **automatic request formulation**

These are longer term R&D issues that the ECS team needs to pursue. The ECS team should show how developments in areas such as these fit into the overall architectural strategy, and allow long term system evolution.

- **Apply domain modeling tools to assure proper system interaction** :
Categorization: ECS, EVO (S/L)

Support systematic evolutionary development of the architecture by developing a domain model that describes the problem-oriented aspects of the architecture that remain relatively constant throughout the life of the system. (Part 3, Ch. 3, pg. 2)

In fact, this has been done through a series of "logical object models" that formed the basis for the ECS System Design Review. The GMU team recommendation should act as a reminder to the ECS team to recheck these domain models periodically to insure their constancy, and to guide detailed design decisions.

- **Location-independent names are mapped into location dependent names to allow ECS objects to be easily moved** :
Categorization: ECS, EVO (S)

The GMU architecture primarily operates with location-independent names, called logical names and mapped names. Logical names provide terms that are meaningful to users, while mapped names are meaningful internally to the ECS Services layer. Special naming and catalog services are used (within the distributed object management layer) to convert mapped names into physical names that correspond to distinct locations. Nowhere within the ECS Services layer are any physical names used.
(Part 3, Ch. 1, pg. 8)

This recommendation is generally consistent with the ECS architecture. Location independence is obtained within ECS through both logical service names and service attributes, which can be employed in service "searches". This approach allows for the evolution of available services and the migration of services in response to system loads or user needs. The GMU team proposes the creation of location-independent names for both logical and mapped services. This is an alternative approach to achieving location independence which is not excluded by the current ECS architecture.

- **Cache user information at client nodes** :
Categorization: ECS, EVO (S)

A user profile manager keeps track of user histories so that users can continue sessions from where they stopped previously, and so that a user's preferred areas of interest can be monitored. (Part 3, Ch. 2, pg. 3)

In the ECS architecture, these features are implemented through stateful sessions between the client and the DataServer / distributed data management system, and through desktop installation of customized or specific services. The user profile parameters that will be captured during a user session have not been fully defined. Furthermore, the degree of session-to-session continuity provided by the system remains to be defined.

- **Cache Global Thesaurus data in a Local Thesaurus at client nodes** :
Categorization: EVO (L)

Portions of the global thesaurus information are cached locally at each user's node. An "optimistic cache coherence protocol" is used to update a local thesaurus if it's information should happen to become out of date. (Part 3, Ch. 2, pg. 4)

These concepts should be integrated into the ECS distributed data dictionary. Because "concept" caching is a bit more difficult than simple term caching, the ECS architecture will need to be flexible enough to evolve to this scenario.

- **Support rules as first-class objects** :

Categorization: R&D

The system must be able to represent rules in such a way that they can be associated with different objects at different times. Therefore, rules must not be represented as attributes of other objects. (Part 3, Ch. 4, pg. 37)

In particular, the ECS team should consider rule-based extensions to their current data dictionary. This will allow for the development of global thesaurus capabilities such as described by the GMU team. The use of condition-action rules in production management has been proposed by the ECS team. The GMU team advocates a more fundamental inclusion of these concepts in the core architecture.

- **Reconfigurable node architecture** :

Categorization: EVO (S/L)

Design the ECS node architecture such that it may be configured in a variety of ways from a standard set of component objects.

This is precisely the reasoning behind the object-based Data Server design of the current ECS architecture. It allows providers to develop as little or as much functionality as they wish, employing "inherited" code where possible in the development of their services.

  - **generated from domain model specification** :

    Categorization: R&D

    GMU proposes the use of their domain modeling tool which combines a standard CASE tool with a knowledge based tool to elicit from the user target system requirements. Ultimately, this could be used to automatically configure an ECS node from a set of standard component modules. (Part 3, Ch. 3, pg. 29)

    While some aspects encompassed by this recommendation are quite "researchy", others are nonetheless more pragmatic. For example, it is conceivable that an Earth Science Data Type library could be developed in such a fashion as to support CASE-based configuration as the GMU team suggests.

- **A variety of information management services are recommended** :

Categorization: ECS, EVO (S/L), R&D

The ECS node architecture is composed of over 15 distinct servers that address a variety of information processing tasks. The functions of each of these servers are described. Some of these servers are well within current technology, while others may require further research before they can be realized in practice. (Part 3, Ch. 2)

The ECS information management services are provided through a common gateway that then maps to the implementation "servers", as described by the GMU team. The sophistication of information management services provided by ECS will progressively expand through the four major releases planned through 2002.

Network Management

- **Use Virtual Client Protocol for reduced network traffic** :

Categorization: ECS, PDR

Reduce network traffic by designing a protocol which allows a secondary server to return its response directly to the primary client, rather than relaying the data through an intermediate requestor.
(Part 3, Ch. 2, pg. 24)

This is envisioned for ECS, as outlined in the ECS end-to-end scenarios presented at SDR. The implementation details will be presented at PDR.

- **Make use of ATM switches** :

Categorization: ECS, EVO (S)

The overall network design should have the wide area network supported with ATM technology. Various tradeoffs must be considered to determine whether ECS should own its network or purchase use from a public carrier.
(Part 6, Ch. 1, pg. 12)

ATM technology is being investigated by both ESDIS and the ECS team as the likely technology for DAAC internetworking. Considerations for more widespread access, including ATM-to-the-desktop, are evolvability concerns as the Internet undergoes infrastructure upgrade. They may uncover programmatic and cost issues as well.

- **Be CORBA compliant (no rationale given)** : (Part 6, Ch. 3, Section 3.8)

Categorization: ECS, EVO (S/L)

The ECS service infrastructure is CORBA-based, with shorter term implementations based on DCE and longer term migration to fully compliant CORBA infrastructures.

- **Manage future consolidation and re-allocation of DAAC resources :**

Categorization: EVO (L), PRG

Network infrastructure should be kept sufficiently flexible so as to accommodate future changes in DAAC responsibilities and processing loads.

In order to support longer term re-allocation of services amongst DAACs, the network infrastructure should be developed in a modular fashion. This would allow reallocation of resources as necessary to accommodate changes in DAAC responsibilities. Programmatic support for this concept needs to be developed consistent with the technical approaches.

- **Be able to exploit idle computing resources on the network** :

Categorization: EVO (L), R&D

Non-ECS sites can register idle CPU cycles with the ECS Product Scheduling Servers so that non-ECS sites can be used to process or re-process standard data products (Part 3, Ch. 2, pg. 28)

In fact, this is addressed in the ECS Conceptual Architecture, where NSF supercomputer center resources are identified as candidates for addressing peak processing needs in response to phenomenological events. The current ECS concept supports the "advertisement" of such resources into ECS. The implementation of these services needs to be addressed.

- **Separation of data and control paths** :

Categorization: ECS

Use an FDDI LAN for control messages and a high speed HIPPI channel for data transfers. (Part 6, Ch. 2, Section 2.2.4)

The importance of this design concept is recognized by the ECS team. E.g., most of the COTS storage and processing products being evaluated by the ECS team employ this design concept.

- ***Network-attached storage*** *:*

Categorization: ECS

Make use of new technology which allows storage device controllers to be connected directly to a network so that data can flow directly from the storage device to the client application without going through the server computer's memory.
(Part 6, Ch. 2, Section 2.2.3)

This recommendation is consistent with current ECS implementation plans.

Other Architectural Issues

- **DADC Concept** (Part 5, Ch. 1)
  - **provide multiple access points to EOSDIS**
  - **integrate EOSDIS with GCDIS under framework of UserDIS**
  - **provide interoperability between different domains**
  - **provide higher level of information management**
  - **support higher level data products**
  - **provide domain-based support services**
  - **provide customized user interfaces**
  - **establish domain communities and domain networks**
  - **manage federations of user communities**
  - **develop common community standards**
  - **provide specialized sales and accounting procedures**
  - **organize domain-specific knowledge**

Categorization: ECS, EVO (S/L), PRG

The DADC concept in general, as well as the specific items listed above, are consistent with the current ECS architecture. Multiple access points are provided through heterogeneous services and interfaces, including direct query interfaces, WWW-based hypertext interfaces, and text-based interfaces. EOSDIS components are being developed with longer term GCDIS and UserDIS in mind. Domain interoperability and higher level information management are provided through the development of DIM (distributed information management) services. Higher level data products are supported through SCFs as data providers and additional, value added providers. Domain-based support services are supported through the existing DAACs, and extended as necessary by SCFs and value-added providers. Customized user interfaces and specialized accounting procedures are also part the this concept. Finally, the concept of a federation of domain communities with domain-specific knowledge, all tied together by common standards is consistent with the ECS concept implemented through Data Server, LIM, and DIM services. The creation of an external DADC-type interface to the ECS is a programmatic issue. Funding for the implementation of DADC organizations is also a programmatic issue.

- **Consolidation of DAACs :**

Categorization: PRG

Many alternatives for the evolution of DAAC's exist. GMU suggests that the evolution to four main DAACs with the others becoming SCF's or DADC's is likely. (Part 5, Ch. 1, pg 10)

The existing DAACs have been selected based on historical support to domain specific users. Changes in the responsibilities of these facilities are a programmatic issue, though the ECS team has identified this as an evolutionary test.

- **Decouple non-earth scientists from direct DAAC access :**

Categorization: EVO (L), PRG

In order to relieve the DAAC's from the potentially tremendous PULL demand from non-science users, it is recommended that intermediate facilities such as DADC's be used to service these users.

The ECS team's Value added provider (VAP) concept is intended to address this concern. Funding for the implementation of VAPs is a programmatic issue.

- ***Coordination of GCDIS with EOSDIS :***

Categorization: EVO (S/L), PRG

GCDIS offers the opportunity for NASA to significantly enhance the value of EOSDIS to the user community. Software developed for EOSDIS may also reduce costs for GCDIS prototype development. (Part 5, Ch. 2)

This was one of the GCDIS "drivers" for EOSDIS, and has been considered in the ECS architecture. This will require continued coordination between the ECS team and the larger GCDIS community to insure adequate functionality for GCDIS.

- **Storage system assessment** (Part 6, Ch. 2)
  - **explore high performance storage systems (HPSS)** :

Categorization: ECS, EVO (S), R&D

Consider use of a scalable distributed file and storage management system that can handle large volumes of data and handle high data transfer rates (Part 6, Section 2. 3.1)

The ECS team has a number of evaluation and prototyping efforts underway and planned for the evaluation of large mass storage systems. The team has also planned on mass storage technology developments that will require archive technology phasing over time.

- **Parallel database management and information retrieval systems**
  - **prototype using Oracle and UniSQL** : (Part 6, Ch. 3, Section 3.8)

Categorization: EVO (S), R&D

Database management systems will evolve in a number of directions, including increased parallelism, and extensions into object oriented implementations. The ECS team is planning for such advances, evaluating available databases as they are available, and integrating them into the architecture at the appropriate time.

- **Thoughts regarding management of "large science" projects** :
Categorization: PRG

The failures of previous large science projects are examined, and new principles for managing large science projects are considered. (Appendix B)

The GMU team's experiences in this area are quite useful. The observations and recommendations of the GMU team have been communicated to NASA managers within the GSFC ESDIS Project Office and the Mission to Planet Earth Program Office at NASA Headquarters.

- ***Analysis of the scalability of the push-pull model*** :
Categorization: ECS, PDR

The properties of PUSH and PULL are examined and compared. (Appendix B, pg. 14)

The GMU team's analyses should be addressed along with ECS team modeling at PDR.

- **Process level 4 data at DADC's, levels 0-2 at DAAC's, and level 3 at both** :
Categorization: PRG

This will keep most of the push functions at the DAAC's while domain-oriented knowledge and data sets reside at DADC's. (Part 5, Ch. 1, pg 5)

The ECS architecture would support such a configuration, although current programmatic assignment places these production responsibilities at the DAACs (levels 0 - 3), the Data Assimilation Office (for level 4 products defined within the current scope) and the Science Computing Facilities(for special products at levels 1 -4). The reallocation of production processing responsibilities to different organizational entities is a programmatic issue.

- **User Scenarios** :
Categorization: ECS, PDR

User scenarios are provided for terrestrial, oceanographic, and meteorological studies. Each scenario has quantitative estimates of the data volumes that will be involved. Also, user access is examined and estimates of intra-arrival times for queries are calculated. (Part 2)

The ECS team should address GMU team scenarios at PDR.

- **Technology Assessment** :
Categorization: ECS, PDR

Technology assessments are provided for ATM, Data Storage, and High Performance DBMS technologies. (Part 6)

The GMU team's technology assessments should be integrated into ECS technology assessments, and addressed at PDR.

- **Lessons from the Human Genome Project** :

Categorization: ECS, PDR

A summary of lessons learned from the Human Genome project is provided. This is of extreme relevance to EOSDIS, because it too is a large-scale data intensive scientific endeavor. While the overall volume of data is much less in the Genome project than it is for EOSDIS, issues of data access and user collaboration are still of prime importance. (Appendix A)

The ECS team should integrate lessons learned from the Human Genome Project, and should find additional projects of relevance that share the characteristics of a large-scale scientific information system.