GENESIS: The General Earth Science Investigation Suite

Thomas Yunck, Brian Wilson, Amy Braverman, Elaine Dobinson and Eric Fetzer
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109

***Abstract* – GENESIS is a NASA-sponsored partnership between the Jet Propulsion Laboratory, academia, and three NASA data centers to develop a new suite of web services tools to facilitate multi-sensor investigations in Earth System Science. Residing within a framework known as SciFlo, these tools will offer versatile operators for data access, subsetting, registration, fusion, compression, and advanced statistical analysis. They will first be deployed in a model server at JPL, and later released as an open-source toolkit to encourage enhancement by independent developers. While the initial work will focus on four premier atmospheric sensors – AIRS, MODIS, MISR, and GPS – the modular design offers ready extension and reuse on many Earth science data sets. The SciFlo design grew out of the pressing needs of scientists active in studies with these new sensors. The tools themselves will be co-developed by atmospheric scientists and information technologists from several institutions. At each step the tools will be tested under fire within active investigations, including cross-comparison of spaceborne climate sensors; cloud spectral analysis; upper troposphere-stratosphere water transport; and global climate model testing. The tools will then be evaluated by our partner data centers and later infused into their operations.**

INTRODUCTION

The General Earth Science Investigation Suite (GENESIS) was selected in 2003 under NASA's REASoN (Research, Education, and Applications Solution Network) program. The GENESIS team includes scientists and information technologists at JPL, UCLA, the University of Maine, Scripps Institution of Oceanography, and three NASA data centers (DAACs). The principal objectives of GENESIS are to alleviate critical data bottlenecks and provide new fusion and analysis tools for multi-sensor Earth System Science. While the tools are designed for reuse across many science disciplines, GENESIS will focus on the needs of NASA's premier atmospheric sensors, including AIRS, MODIS, and MISR, on NASA's Terra and Aqua spacecraft, and the GPS occultation sensors on CHAMP, SAC-C, and GRACE. The Langley, Goddard, and JPL (Physical Oceanography) DAACs join this effort to provide the data products, evaluate key technologies, serve as test-beds, and eventually integrate proven functions into their operations.

Table I presents key GENESIS objectives over the next four years. To approach this we've assembled a team of scientists active in investigations with the target instruments, together with IT specialists. We began by formulating a set of multi-sensor science investigations and calibration/validation scenarios central to NASA's Earth science priorities. We then worked through these scenarios and performed selected tests in data manipulation with current NASA archives to identify critical obstacles. These proved plentiful – data access, subsetting, fusion – and illustrate why so little multi-sensor Earth "system" science is done today.

TABLE I
GENESIS OBJECTIVES

- Provide easy web-based access to science products from AIRS, MODIS, MISR, and GPS
- Co-register products on a common global grid
- Enable swift, versatile subsetting of all products
- Provide advanced fusion tools and statistical summarization and data mining operators
- Create a model server to stage and operate upon the collected products
- Provide tools to create and deliver on-demand, user-specified, custom products and operators
- Test, refine, and apply these tools in a diversity of real science applications
- Release the system publicly in an open-source modular toolkit designed for easy reuse
- Deploy in the JPL, GSFC, and Langley DAACs

We then sketched the elements of an efficient, distributed atmospheric data information system. Finally, we distilled the major IT problems cutting across user scenarios with a view to evolving today's infrastructure towards the ideal. We sought out strategies that: (1) make thorough use of existing DAAC services; (2) can be readily infused to enhance those services; and (3) bring together disparate science products within a common framework. The result is the novel SciFlo web services architecture. Where possible we have taken proven solutions – the "web services" paradigm – rather than inventing new ones, and tailored them in novel ways to the demands of NASA's Earth science data systems.

RELEVANCE TO NASA'S EARTH SCIENCE OBJECTIVES

Current NASA Earth science priorities strongly emphasize weather and climate. Of six research focus areas cited in the most recent NASA Earth Science Strategic Plan [1], five directly concern weather and climate:

- Weather prediction
- Water and energy cycle
- Carbon cycle and ecosystems
- Atmospheric composition
- Climate variability and change

AIRS, MODIS, MISR, and GPS are central to these efforts and to NASA's larger ambition to characterize, understand, and predict Earth's behavior. The Earth Observing System, conceived nearly 20 years ago, introduced to the world the notion of Earth System Science (ESS) – the study of Earth as a cou-

pled web of physical processes and feedbacks. Today EOS is returning a flood of new data – a volume approaching three Terabytes every day. Yet the promise of ESS is still pending. Owing in part to serious obstacles in obtaining and subduing these diverse products, little multi-sensor "system" science is yet being done. The GENESIS tools will help to inaugurate Earth System Science and will advance a modern data system architecture for realizing the broader vision of NASA's Earth Science Enterprise.

Instrument Summaries

Summaries of the four instruments – AIRS, MODIS, MISR, and GPS – that are the focus of GENESIS are given below.

AIRS – The Atmospheric Infrared Sounder (Fig. 1) flying on NASA's Aqua spacecraft performs multi- and hyper-spectral sounding of the atmosphere and the surface, resolving 2380 channels within the 3.7–15.4 μm infrared band. The sensor scans ±49° cross-track through nadir, with a scan period of about 2.7 sec. With this wide swath width, AIRS can cover nearly the entire earth every day. The "AIRS suite" (including an Advanced Microwave Sounding Unit and a Humidity Sounder for Brazil) generates profiles of atmospheric temperature and moisture and measurements of precipitable water, surface temperature, cloud fraction, cloud top height, and total atmospheric ozone. The temperature profiles are accurate to about 1 K and have a vertical resolution of about 1 km. Horizontal resolution at the surface is 15 km. In all, AIRS acquires acquires eight gigabytes of raw data each day.

MISR – The Multi-angle Imaging Spectro-Radiometer (Fig. 2) flying on NASA's Terra spacecraft operates largely at visible wavelengths, measuring reflected sunlight in four color bands (blue, green, red, and near-IR) and at nine distinct viewing angles at once. With a considerably narrower swath width than AIRS, MISR takes about ten days to cover the entire earth. The MISR data can be used to distinguish different types of atmospheric aerosols, cloud forms and cloud cover, and land surface covers. These help to illuminate the division of energy and carbon between the land and the atmosphere, and to explore the effects of aerosols and clouds on climate. With the aid of stereoscopic techniques, MISR enables construction of 3-D models and estimation of the total sunlight reflected from different environments. MISR achieves a surface resolution of 275 m and returns about 30 GB of data each day.

MODIS – The Moderate Resolution Imaging Spectrometer flying on both Terra and Aqua (Fig. 3) operates over a broader frequency band than AIRS, measuring radiances over 0.4-14.4 μm, but resolves that band more coarsely, into just 36 channels. It achieves horizontal resolutions of 250-1000 m, depending on the band. Products include the boundaries and various properties of clouds, aerosols, and land; ocean color; atmospheric moisture and temperature profiles; surface and cloud temperatures; cloud top altitude; and total column ozone. MODIS returns about 60 GB of raw data each day.

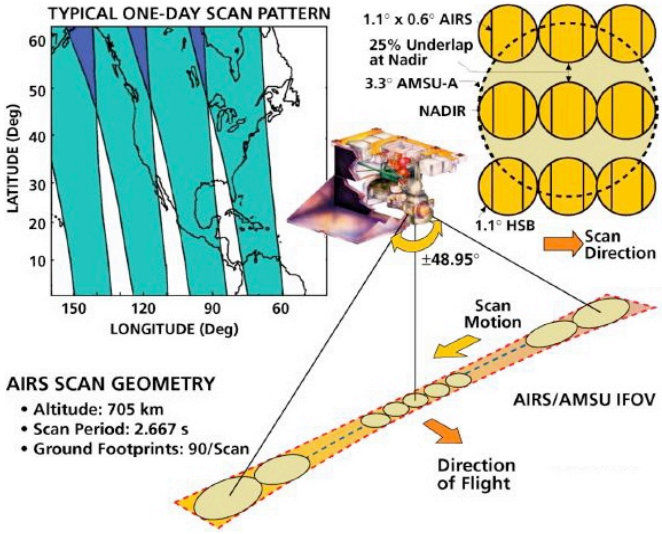

Fig. 1. AIRS instrument and mission overview.

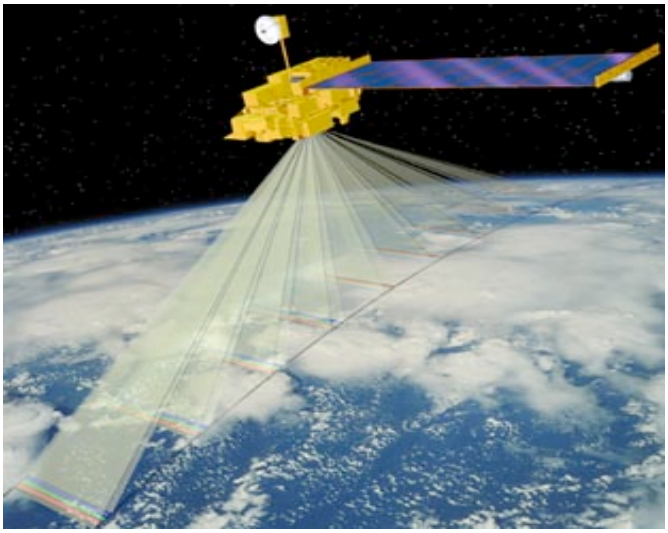

Fig. 2. MISR observing from nine simultaneous look angles.

Fig. 3. The MODIS Instrument flying on TERRA and AQUA.

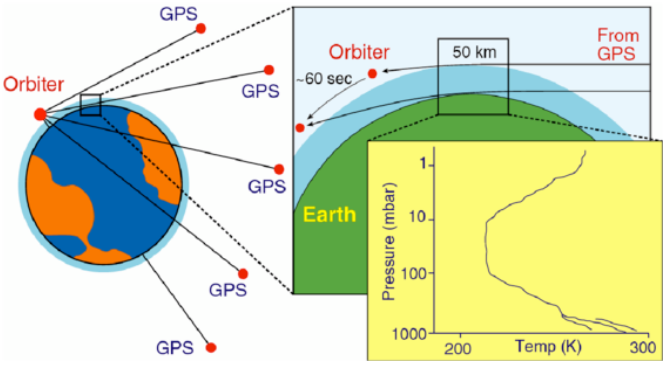

Fig. 4. Sketch of GPS occultation.

GPS – In contrast to spectrometers, which passively observe emissions from the atmosphere and surface and measure radiance in different frequency bands, GPS occultation observes active radio signals and measures the changing path delay of the signal passing through the atmosphere (Fig. 4). Each profile is virtually self-calibrating, beginning or ending with a zero measure in free space. Products include atmospheric refractivity, pressure, temperature, and moisture (below ~5 km). Because the raypath is precisely known, GPS (uniquely) recovers true geopotential heights to <10 m, which yield global pressure contours and geostrophic wind fields. Recent results indicate that single profiles are accurate to about 0.5 K, and multiple profiles average to better than 0.1 K between 5 and 30 km – ten times better than other techniques [2]. Vertical resolution is about 100 m near the surface, falling to about 1 km in the stratosphere. A single receiver can obtain about 600 profiles/day globally, returning about 40 MB of data each day.

Science Foundations

The GENESIS team has defined six foundation science scenarios upon which to design, refine, and demonstrate our data technologies. These form a representative set of the limitless possibilities and fall into three categories:

1. Sensor cross-validation and calibration:
   – Calibration of spectrometers with GPS
   – AIRS/MODIS cross-instrument validation
2. Focused climate process studies:
   – AIRS/MODIS cloud spectral analysis
   – Troposphere-stratosphere water transport
   – Exploring the aerosol "indirect cloud effect"
3. Climate model validation/improvement:
   – Global testing of cloud models

The original REASoN solicitation observed that "Perhaps the greatest roadblock to fundamental advances in our understanding of climate variability and climate change is the lack of robust and unbiased long-term global observations." Because climate signals are subtle (~0.1-0.3 K change per decade), current sensors require meticulous calibration of drifting biases. The REASoN solicitation noted that for climate monitoring "the focus is on…construction of consistent datasets from multi-instrument, multi-platform, and…multi-year observations with careful attention to calibration and validation over the lifetime of the measurement." The first two scenarios address issues of multi-instrument/platform/year calibration and validation. These three categories immediately suggest a fourth:

4. Climate monitoring and signal detection

Detecting climate signals demands exceptionally precise and stable data. GPS calibration will improve long-term temperature stability of EOS sensors to about 0.05 K from the currently specified 1 K, transforming AIRS/MODIS/MISR into powerful climate sensors. From the first year we will begin to lay down an absolute data record that can expose climate signals within a matter of years rather than decades.

Technology Obstacles

With present data systems such studies are burdensome and in many cases impractical. Each requires swift access to, selection from, fusion of, and operation upon large volumes of incommensurate products from multiple sensors and archives. This invariably requires custom code and deep expertise to accomplish properly.

GENESIS will create new tools to facilitate multi-sensor science; exercise them in real scientific investigations; deliver them in an open source toolkit that automates many steps in the process; adopt uniform and flexible standards; and provide a model archive of multi-sensor data, co-registered and cross-validated. Here we describe several of today's persistent IT bottlenecks and briefly outline our vision to address them.

Data Volume and Access. A single-instrument dataset may contain thousands of granules and terabytes of data. A typical AIRS, MODIS, or MISR swath is broken into many granules, each containing tens or hundreds of parameters. Assembling a regional dataset for a specified interval involves subsetting the granules for time and location, retrieving the parameters, and aggregating the results. Scientists often simply retrieve a full global dataset, taxing network bandwidth, then subset and aggregate them locally with custom code – a demanding chore for which many investigators must employ specialists. Within SciFlo, the request/response cycle will be fully automated and custom functionality migrated to the DAACs.

Time/Location and Parameter Subsetting. Snags in finding and retrieving data through the EOS Data Gateway and elsewhere have prompted the DAACs to begin supporting time/location and parameter subsetting. The GSFC DAAC will soon support both synchronous (online) and asynchronous access with versatile subsetting. SciFlo will exploit two of these tools that employ wu-ftp and Web Coverage Service (WCS) protocols and serve the large stores of on-line data in the Data Pools. Current GSFC tools emphasize grid data. SciFlo will provide swath subsetting operators and a framework in which user-designed subsetting can be tested locally, then transparently deployed at a DAAC. Any program that reads an HDF file and writes a new one can be transparently used as a data cutter.

Variety of data formats. The variety of data formats in Earth science is enormous: HDF, CDF, netCDF, GRIB, GeoTiff, XML, etc. Since HDF-EOS is the only widely used format that handles both swath and grid data, SciFlo will adopt it as a standard container to structure and transport data, though it will also handle HDF and netCDF files and other containers for regular grid data. Many scientists have learned to apply these formats with higher-level API's such as those in the Interactive Data Language (IDL). SciFlo will offer some operators in Fortran or C/C++ for high performance but IDL is sufficient for many. We will also employ Earth Science Markup Language (ESML) readers; operators created with ESML can transparently handle multiple formats. SciFlo services will be offered via a Web Coverage Server that converts to netCDF, GeoTIFF, and other formats, a function already hosted in the JPL DAAC Ocean ESIP Tool [http://podaac-esip. jpl.nasa.gov/poet/].

Coincidence searching. Finding overlaps between MODIS swaths from Terra and AIRS swaths from Aqua, or between AIRS and GPS, is nontrivial. To our knowledge there are no services today that solve this random access lookup-and-intersect problem in a robust way for large EOS datasets. GENESIS is developing a coincidence searching service for EOS swath and point data. SciFlo will tap that effort to find swath overlaps for sensors on multiple platforms.

GENESIS previously developed a server, middleware, and web interface to provide flexible access to GPS products. The interface consists of web forms and a Java application to subset and visualize profiles and grids [http://infolab.usc.edu/GENESIS/index.html]. The middleware is a Java server that provides web interfaces, talks to the Java application, and connects to an Informix object-relational database. ESRI and Geodetic Spatial Datablades facilitate queries based on *<lat, long, altitude, time>*. A spatial index structure, R-tree, expedites queries. The same method will be used for on-demand coincidence searching for GPS, radiosonde, and AIRS data. We have installed a SOAP service that allows one to submit an arbitrary SQL query to any of our Informix databases; coincidence searching services will be layered on top of that capability.

Data Fusion. With overlaps in hand co-registration and data fusion can proceed at various levels of complexity: We can simply overlay one set on another for visual inspection; or we can interpolate or re-project data to a common grid or projection – facilities common in GIS packages (e.g., ARC INFO, GRASS). More advanced techniques exploit spatio-temporal relationships in the data to optimally interpolate to common locations [3], [4]. SciFlo will feature reusable operators implementing a range of methods including advanced statistical fusion of data from multiple sensors. We will exploit existing reprojection software such as HDF-EOS to GeoTIFF (HEG) and the MODIS Swath-to-Grid Toolkit (MS2GT).

Multivariate Statistics. To estimate quantities reliably from multivariate data one must represent their full joint distribution. Consider a set of 10 parameters accumulated over time onto a 1°x1° global grid. The vastness of the 10-dimensional multivariate distribution presents a nearly paralyzing obstacle. One therefore usually saves only a small slice of the available information, such as individual means and variances or covariances of selected parameter pairs. This precludes analysis of the full joint distribution, inhibiting discovery of unexpected connections. SciFlo will offer new tools based on clustering and Principal Components Analysis to summarize multivariate distributions to a far greater level of detail and to compute arbitrary quantities from them.

Software Reuse. While progress has been made in this arena (e.g., reusable API's and classes in object-oriented languages), the methods tie one to a few coding languages. The advent of loosely-coupled distributed computing based on SOAP (Simple Object Access Protocol) remote procedure calls (RPC) has triggered a new paradigm in which remote programs converse through XML. Each operation in a processing stream can become a SOAP-callable web service regardless of the implementation language. The process flow can be specified in an XML document (declarative programming) defining how to assemble the building blocks (reusable operators). SciFlo will provide a lightweight, open source toolkit allowing any executable program with documented command-line inputs to be exposed as a SOAP web service and reused as an operator in processing flows. By combining distributed computing with simple declarative programming, SciFlo will raise the bar on software reuse, and will easily support such widely recognized standards such as ESML, WCS, WMS, DODS, and the Open Data Access Protocol (OpenDAP).

Automated Processing of Structured Data. The first generation of web software was tied to presentation of unstructured data in HTML, and to a stateless http protocol that returns unstructured results in HTML. While these have been adapted to large binary datasets (e.g., DODS; WMS/WCS), they are not adequate for large-scale automated data processing. The second generation is moving toward the automated manipulation of semi-structured and structured data represented in XML, and to a paradigm in which automated programs communicate with one another asynchronously in XML. SOAP and XML have now become flexible enough, powerful enough, and fast enough to form the "glue" for an entire data processing system. The challenge is to adapt the emerging web service standards to the needs of scientific data processing: vast datasets, binary formats, and complexity of metadata.

Adoption. Every new technology must vie for acceptance. SciFlo confronts this with a lightweight, language-independent framework; loosely-coupled distributed computing; simple declarative programming; end-to-end processing flows expressed in XML; an execution engine exploiting parallelism at many levels; flexible subsetting, co-registration, and statistics operators; standard operator and data types specified by XML schemas; reuse of DODS, WMS/WCS, and ESML standards; open source software; and one-step installation on Linux clusters. Use of SciFlo will not require programming in the usual sense; the only "API" consists of writing XML documents with an outline-based XML editor.

TECHNICAL APPROACH

A major challenge facing NASA data centers is the generation of subsets and custom products delivered over the web. Since the centers cannot anticipate all requests, they have a need for smart subsetting and custom summarization. Ideally, users would specify a remote data subset from within an application (browser, IDL, Matlab) and receive the results immediately, directly in the application. DODS and WCS are used widely because they can be called inside IDL and Matlab (DODS) or a browser (WCS). Many users, however, would like to customize complex operators to suit particular needs.

Two developments in network computing are of particular interest: (1) tightly-coupled Grid Computing in which software toolkits link distributed clusters or supercomputers, and (2) loosely-coupled distributed computing, or "web services," in which protocols like SOAP call procedures remotely over the net and return results.

SOAP web services were originally created for business ("e-commerce") use; the SOAP Web Service Definition Language (WSDL) and Universal Description, Discovery, and Integration (UDDI) standards are driven by business needs. SOAP offers a standard for providing and consuming web services; WSDL provides a way to precisely describe each service; and UDDI provides a means to publish and discover services in a searchable catalog. Web services that communicate in XML can be coded in a mix of languages (perl, python, Java, C, C++, C#). A structured web service can be thought of as an XML document in WSDL, and many software toolkits automatically generate code from the WSDL description.

Web services are now being applied in science. The Globus toolkit for Grid Computing will migrate toward the Open Grid Services Architecture (OGSA) [5] in which most functions (e.g., discovery, authentication, security, submission, monitoring, and steering) will be recast as web services. Only tightly coupled services involving execution of numerical algorithms on a grid will remain in the old paradigm.

A. Enabling Technologies for SciFlo

SciFlo combines four core ideas to promote software reuse and create a marketplace for science analysis services: loosely-coupled distributed computing using SOAP; exposing scientific analysis operators as SOAP web services; specifying a data processing stream as an XML document; and a dataflow execution engine for a parallel execution plan and load balancing.

Loosely-coupled distributed computing. In the SOAP and XML-RPC protocols, remote procedures are invoked through XML messages, without reference to implementation details. Other distributed computing standards (CORBA, DCOM, Java RMI) can be problematic as they are more tightly coupled and have a fairly steep learning curve. Java RMI simplifies matters but ties one to Java, which is ill-suited to intensive computing. In contrast, SOAP is lightweight, language-independent, and ideal for loosely-coupled distributed computing.

Analysis operators as SOAP web services. Scientific operators are often intricate, with many configuration parameters, and may involve vast and complex inputs and outputs. For example, Conquest (Concurrent Queries over Space and Time), developed at UCLA [6], decomposes queries consisting of fine-grained operators, optimizes, and then executes them in parallel. The focus of SciFlo is coupling medium-grained operators, and components that can be local or remote, in a simple, declarative manner; a SciFlo operator accepts one or more complex inputs, performs tasks tailored with configuration parameters, and yields one or more complex outputs. Inputs and outputs can be local HDF files and metadata, pointers to remote HDF files or array slices, or the actual numbers.

Power of declarative programming. A key goal of SciFlo is to provide XML standards that describe analysis operators (verbs) and groups of operations (processing flows) to serve as building blocks in an arbitrary processing flow. By encapsulating code within well-defined operator interfaces described in XML, new programs can be assembled with a simple XML document. The user "declares" the new processing flow and the execution engine does the rest. SciFlo will provide the "glue" allowing any user to expose analysis operations as SOAP-callable web services and to invoke remote operations as part of a data processing flow without writing new code.

There is much activity in the commercial sector to coordinate and assemble web services for business. IBM offered a Web Services Flow Language, superseded by the Business Process Execution Language for Web Services (BPEL4WS), to describe business processes. Two auxiliary specifications, WS-Coordination and WS-Transaction, have appeared, and a new Web Service Choreography Interface. We have chosen not to use these specifications because the designs are evolving, tailored for business, and require proprietary editing tools. SciFlo will adopt WSDL to describe operators and UDDI to catalog and discover them, but specialized for science: compatible with WCS and DODS, employing simple XML flows, and offering a substrate with hooks for semantic web activities in the science realm.

Dataflow Execution Engine. At the core of systems like Conquest and Sciflo are dataflow execution engines. SciFlo carries the idea further by engaging open, interoperable web services with on-demand integration of operators. It contains a dataflow engine that parses the XML description of the process flow, creates an optimized execution plan, distributes and load balances the computation, parallelizes if possible, and coordinates the execution of each operator. Operators can be local executables or scripts, or remote entities that require the engine to invoke a remote SOAP service, make a WMS/WCS call, submit an http POST request, or submit a one-line http GET request (e.g. a DODS URL or a GrADS-DODS server call). The engine creates the code to move data inputs and outputs between nodes, sequence operations, execute operators in parallel, update a status log, and deliver results.

SciFlo and Conquest represent somewhat different computing paradigms. Conquest balances flexibility and performance,

automatically parallelizes, migrates operators, and requires operators in C++. SciFlo emphasizes flexibility and ease; automatically parallelizes; supports discovery, migration, and automated installation of executables; and is language-independent. Though each operator can be a native executable for high performance, one would not generally elect to do intensive computing as a complex SciFlo operation.

B. Features of SciFlo

Here we sketch the SciFlo design and its key technologies. We plan to build two versions: an operational prototype with a subset of features, to be exercised in several science investigations, followed by a full-function, operational system.

Hardware Paradigm. A SciFlo server will be deployed on a local cluster – Unix stations, Linux PC's, rack-mounted server blades – at the host site, and link with remote SciFlo servers across the web, presenting a broad mix of latencies.

Streaming Data Between Operators. The data transfer method determines the granularity of operators that can be executed efficiently. In a tightly-coupled cluster, in-memory transfer can be used. For loosely-coupled computing, small objects can be sent in binary or XML formats and larger objects in files by ftp. SciFlo occupies the loosely-coupled domain supporting large and small objects with three modes of transfer:

- Large – via ftp and DODS URL's, or (locally) via shared disk space;
- Medium – via extended DODS URL's and the OpenDAP protocol;
- Small – in XML text within a flow document.

Data and Operator Types. SciFlo will *name* objects hierarchically and use XML schemas to *describe* types. The Earth Science Markup Language (ESML) is an XML specification for syntactic metadata (structure, type, format) indicating what each variable represents, and other "content" metadata for discovery. SciFlow will adopt existing ESML descriptions and create new ones. We will also provide a placeholder for "kind" data to allow for later semantic web advances. Each data object and operator will be labeled with both a *type* and a *kind*. Type and kind elements can be formally represented in a semantic ontology using the Resource Description Framework (RDF).

Names will be assigned with root 'sfl' (SciFlo). An AIRS granule containing all Level 2 products, for example, would be type 'sfl.data.airs.l2.granule' while a MODIS temperature granule would be type 'sfl.data.modis.l2.temperatureGranule'. An operator to co-register data from two swaths would be type 'sfl.op.coreg.swath2swath'. The HDF file type will be determined both by the name and an XML schema describing variables and attributes.

Parallel Computing. The SciFlo server will apply parallelism at many levels:

- Within a single operator
- With multiple processes on a single node
- With multiple threads within a forked process to execute, monitor, and respond to queries
- With multiple processes created by launching operators or sub-flows on local nodes
- By invoking a remote operator via a SOAP or WCS call and waiting for the results
- By redirecting a flow or sub-flow to a server that can access a data source locally or that can efficiently execute a CPU-intensive operator
- By partially evaluating a flow and then redirecting
- By invoking a flow multiple times if the source operator(s) yield multiple object/files that need to be processed (implicit file parallelism)

Flow Execution. Based on the expected execution time and computing resources described in the flow specification, flows will be executed in either immediate or queued mode. If the immediate mode is specified and the run time exceeds a threshhold in the server configuration file, the flow may be aborted. The execution process consists of:

- *arrival* – flow execution request arrives at server
- *syntax check* – see if document is well-formed
- *queue* – push the flow onto an execution queue
- *dequeue* – move the flow to the execution phase
- *parse* – validate document against XML schema
- *type check* – verify that operator input/output data will be of the correct types
- *embellish* – insert implicit unit or format conversions to fill in missing steps
- *plan* – determine the execution plan and annotate the flow document accordingly
- *execute* – execute parallel flow according to plan
- *deliver* – deliver results to the requester

At first, 'embellish' will supply minor missing steps using unit and format conversion operators. Later, it will be possible to leave out more substantial steps and have them automatically filled with the "best" operator that can be discovered in the catalog. The flow author can also specify conversion operators that should be used to fill gaps. The user will receive the embellished flow document. Delivery consists in returning results in XML to the SciFlo client or in HTML to a browser. Results will include intermediate and final data objects/files, the embellished flow document, and an execution log.

Server Operation. The execution engine comprises separate processes, executing on one node or many, that talk via SOAP; an XML file will specify the node configuration to employ. The server is a SOAP service and can register itself in a UDDI catalog; one could then use UDDI discovery to find SciFlo nodes to combine. Thus, SciFlo will function as an adaptive parallel server that can be reconfigured by editing XML files.

Fig. 5 depicts the SciFlo master server and its links to other servers. Front-end nodes field requests, check them, and push them to a queue. A master node retrieves the flow and executes

it through the planning phase. If the plan calls for flow redirection, the annotated flow document will be sent to the proper server. If the flow starts locally, parallel execution will begin on a master server and may tap multiple CPU's. The figure shows several remote operators executing, starting at the JPL DAAC.

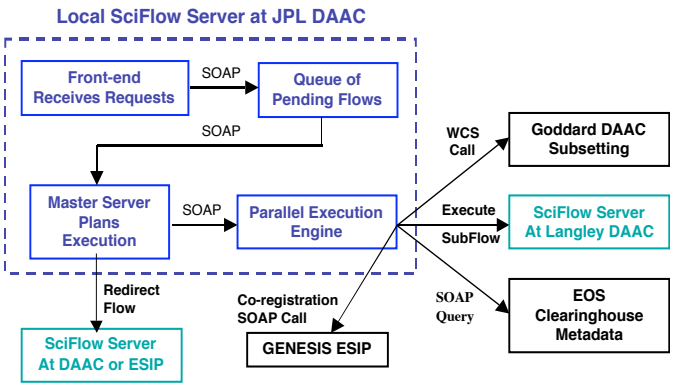

Fig. 5. Flowchart of SciFlo distributed computing.

USAGE SCENARIOS

We envisage SciFlo being used in at least three kinds of scenarios: large-scale automated data processing; as a toolkit to create custom services; and for scientific discovery. The first has been amply illustrated; here we discuss the other two.

Software Toolkit. To illustrate the power of combining operators, consider constructing a service for swath granules that provides time, location, and parameter subsetting, and aggregates the product over time. This can be realized in four steps: (1) find granules within the time range, (2) intersect each granule with the lat/lon/alt region, (3) subset the surviving granules, and (4) aggregate them over time. Each step can be captured in a reusable operator; new subsetters can then be built by assembling the blocks in a variety of flows. A scientist could construct a custom subset and aggregation operation and call it from a browser with a WCS-like URL. This way SciFlo can build hundreds of custom services with a similar interface.

Scientific Discovery. Example: Dolores creates a SciFlo by completing a web form, uses a web page to install it as a SOAP service and gets back a link to call it with. She receives the output file, then switches to IDL to use the DODS URL to retrieve results of interest. (Or, she can use a flow with an in-line IDL program.) She tells Waldo about this service and he surfs in, changing parameters at will for variant studies. One missing element is an ability to invoke the analysis within IDL. As the SOAP revolution grows, IDL, Matlab, and their ilk will add SOAP calls. When that hole is plugged, the informed scientist will be able to do it all from IDL or Matlab.

DATA MINING BY SUMMARIZATION

Two challenges posed by EOS data are their volume and multivariate complexity – i.e., the intricate relationships among parameters as they evolve in time and space. Beyond the practical problems of managing such data, there are analytical issues: How can we combine data from different sources? How can one uncover unexpected relationships? For mixed datasets a key question is, What are the hidden connections and how do they evolve? A challenge for the scientist is to frame such questions as testable hypotheses about features of multivariate distributions. This requires tractable representations that preserve the features. To that end GENESIS will:

- Create a summarization tool that preserves approximate multivariate distributions. A "quantization" method used in signal processing yields summarized data that can be analyzed as if they were the original products, with little loss of fidelity.
- Generate summarized data products organized by source, including MISR cloud properties, MISR aerosol properties, and AIRS moisture and temperature profiles and cloud fraction. Each product is an estimate of the joint data distribution for each 1°x1° spatial cell, over a specified interval.
- Create SciFlo operators to build dynamic, user-defined summary datasets from static, precomputed base summary products. For example, a user wanting to test whether some non-linear function of AIRS moisture at several altitudes can help predict cloud fraction could execute a non-linear regression operator and obtain only that result. To rerun variations, the user may prefer simply to order the base product.
- Create SciFlo operators for combining summary products from different sources. For this we will examine several paradigms: traditional methods of matching or interpolating nearby observations; geo-statistical methods, as in [3]; and new methods for inferring joint distributions given only the relevant marginal distributions, as in [7] and [8].

A. Summarization Applications

Summarized data allow scientists to estimate arbitrary multivariate functions and attach realistic uncertainties to them. Traditional Level 3 products provide only the mean and variance of individual parameters at coarse spatio-temporal resolution, limiting potential investigations. Some analysts also provide covariances, correlations, or regressions between selected parameters pairs – useful but still incomplete. Where linear regressions are given, information on relationships that are non-linear, or among three or more quantities, or between unselected pairs is lost. Moreover, realistic information about the data distribution is required for accurate uncertainty measures.

Suppose we wish to estimate the fraction of cloudy pixels of a given type in one grid cell. Assignment of pixels to cloud type is generally made from cloud-top pressure and optical thickness data. With just the means and variances, one would have to assign all pixels in a cell to a single class represented by the mean. One needs the distribution to describe within-grid variation. In [9] we compared the fraction of deep convection clouds estimated from the original data (160 MB) to that from summarized data reduced by more than x1000 (145 KB), obtaining nearly very similar results (Fig. 6).

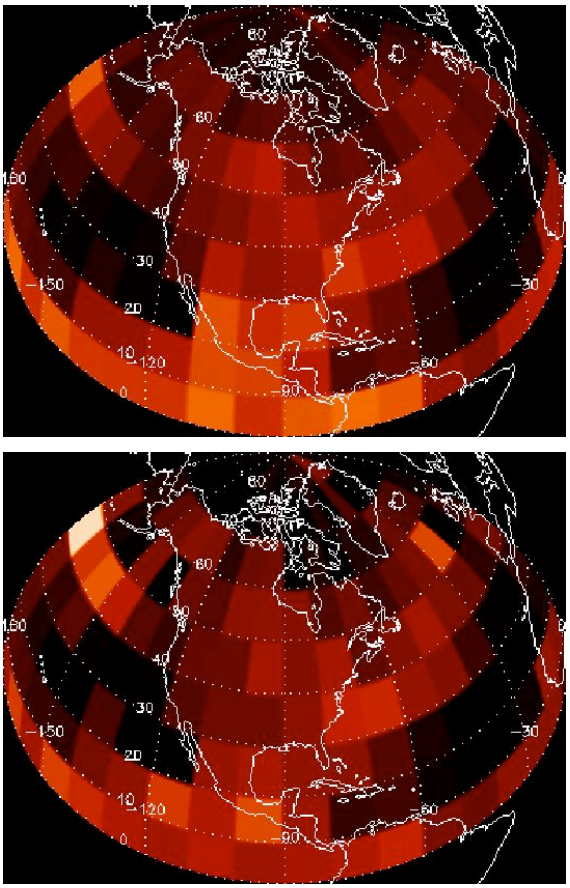

Fig. 6. Fraction of deep convection clouds using ISCCP data from July 1991 from raw (top; 160 MB) and summarized (bottom; 145 KB) data.

B. Aspects of Data Summarization

We wish (say) to summarize the multivariate distribution of d parameters over some time period on a 1°x1° global grid. We partition the data into geographic cells, each holding multiple observations on d parameters. Instead of reporting d means and variances or other traditional summary statistics, we report a small set of representative d-vectors with associated weights and errors for each cell. Each representative stands in for a number of original d-tuples, called a cluster, where the cluster weight is the number of members it contains. Cluster representatives are the centroids of the cluster members, and the error, or distortion, is the mean squared Euclidean distance between cluster members and their representatives. The number of clusters may vary from cell to cell according to the data complexity. If data in one cell are homogeneous, a single representative may suffice. The collection of all representatives, weights, and errors is a quantized version of the original data, with which one can then directly compute as if using raw data. The distortion must be propagated through these calculations to estimate the "summarization error."

The product can be thought of as a kind of multivariate histogram in which the shapes, sizes, and number of bins adapt to the shape and complexity of the data. Unlike ordinary multivariate histograms, which have fixed rectangular bins, each represented by its mid-point, the summary product uses irregular bins and bin centroids to minimize distortion. Moreover, the assignment of data points to bins is determined by the complexity of data in the grid cell, and uses the fewest bins necessary to describe the data.

C. Algorithms

Quantization algorithms, as described in [10], group data points into clusters and compute a single representative for each in a way that balances information loss and compactness [11], [12]. This is similar to statistical clustering methods, such as the k-means procedure [13]. The basic algorithm is:

- Choose k, an initial number of clusters
- Randomly assign each original d-dimensional data point to one of the clusters
- Compute the cluster centroids and counts
- Reassign each original data point to the nearest cluster using some metric
- Update the centroids and counts
- Iterate steps 4 and 5 until convergence

At convergence, the centroids are the representatives and the counts are the weights. From the final assignments we report the mean squared distance between the original points and their representatives as a measure of error for each cluster. Different distance measures give different results. With the ordinary Euclidean metric we have the k-means procedure, always obtain k representatives, and distortion is minimized. With the Euclidian metric plus a penalty for low cluster count, we have the entropy-constrained vector quantization (ECVQ) method [14]. This causes the final assignments to balance compression and distortion. Further details are given in [9] and [15].

Looking Ahead

When GENESIS and SciFlo are operational in three or four years they will present to the user a convenient graphical interface for describing and executing investigations in Earth System Science using data from the AIRS, MODIS, MISR, and GPS instruments. While the details of this interface are yet to be specified, a working concept is illustrated in Fig. 7. This concept is modeled on the layout for the "iMovie" application that comes with Macintosh computers. This is a useful model since laying out the steps of an end-to-end science investigation is much like laying out the scenes of a movie. With this interface, the user will begin by creating a "New Project." The first step will be to describe the research scenario in the "Setup" step, selecting from a pallet of options and entering qualifiers into a form, as needed. The analyst then defines the measurements to be used, again selecting from a pallet. Subsequent steps include Customizing and Summarizing the measurements, Visualizing, Analyzing, and finally Archiving the results, all by selecting from a pallet of operators and entering qualifying parameters, as needed. The resulting "visual program," shown in the bottom strip of the windows in Fig. 7, can be executed step by step for debugging, or submitted as a batch. The finished program can then be exposed as a web service for use by other investigators.

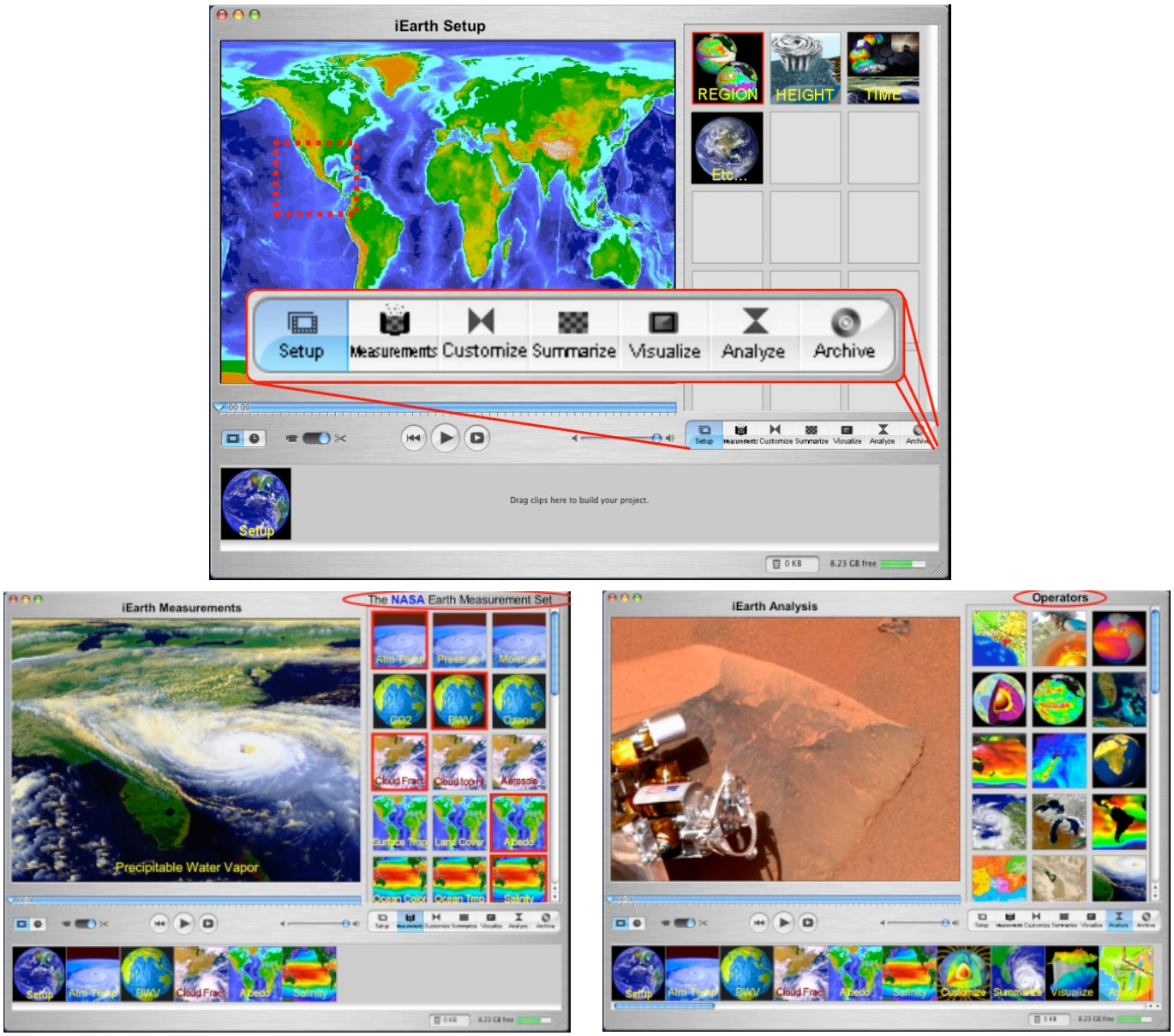

Fig. 7. Mockup of a graphical interface for describing and executing end-to-end investigations in Earth system science with data from diverse remote sensing instruments. Steps include setup to describe the problem parameters (top), defining the measurements (left), and performing various customizing, summarizing, visualizing, and analysis tasks (right). The result is a visual program (bottom strip) defining the full investigation.

ACKNOWLEDGMENT

The work described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

REFERENCES

[1] National Aeronautics and Space Administration, "Earth Science Enterprise Strategy," NP-2003-10-318-HQ, Oct 2003, 86 pp. (http://www.nasa.gov).

[2] Hajj, G.A., C.O. Ao, B.A. Iijima, D. Kuang, E.R. Kursinski, A.M. Manucci, T.K. Meehan, L.J. Romans, M. de la Torre Juarez and T.P. Yunck, CHAMP and SAC-C atmospheric occultation results and intercomparisons, *J. Geophys. Res.,* 109, D06109, doi:10.1029/2003JD003909, 2004.

[3] Cressie, N.A.C. "Statistics for spatial data", Wiley-Interscience, New York, 1993, 900pp.

[4] Kamberova, G., R. Mandelbaum, M. Mintz and R. Bajcsy, Decision-theoretic approach to robust fusion of location data, *J. Franklin Institute*, 336B, 2, 1997.

[5] Foster et al., The physiology of the Grid: An Open Grid Services architecture for distributed systems integration; also, Open Grid Service Infrastructure WG, Global Grid Forum, 22 Jun 2002. (http://www.globus.org/research/papers/ogsa.pdf).

[6] Shek, E.C., R.R. Muntz, E. Mesrobian and K. Ng, Scalable exploratory data mining of distributed geoscientific data, *Second Intl. Conference on Knowledge Discovery and Data Mining*, Portland, OR, Aug 1996.

[7] Dall'Aglio, G., S. Kotz and G. Salinetti, "Advances in probability distributions with given marginals," Kluwer AP, Boston, 1991, 236 pp.
[8] Han, T.S., Hypothesis testing with multiterminal data compression, *IEEE Trans. Information Theory*, IT-33(6), 759-772, 1987.
[9] Braverman, A., Compressing massive geophysical datasets using vector quantization, *J. Computational and Graphical Statistics*, 11(1), 44-62, 2002.
[10] Gersho, A. and R. Gray, "Vector quantization and signal compression," Kluwer AP, Boston, 1991, 760 pp.
[11] Ash, R., "Information Theory", Dover, New York, 1965, 339 pp.
[12] Gray, R., "Source coding theory," Kluwer AP, Boston, 1989, 209 pp.
[13] MacQueen, J.B., Some methods for classification and analysis of multivariate observations, *Proc. Fifth Berkeley Symp on Mathematical Statistics and Probability*, 1, 281-296, 1967.
[14] Chou, P.A., T. Lookabaugh and R. Gray, Entropy-constrained vector quantization, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37, 31-42, 1989.
[15] Braverman, A. and L. DiGirolamo, MISR global data products: a new approach, *IEEE Transactions on Geoscience and Remote Sensing*, 40(7), 1626-1639, 2002.