

Overview of the Tropospheric Emission Spectrometer Ground System¹

Steve Larson
Reinhard Beer
Jet Propulsion Laboratory
4800 Oak Grove Dr.
Pasadena, CA 91109
818-354-0679

Steve.Larson@jpl.nasa.gov
Reinhard.Beer@jpl.nasa.gov

Final
10/20/00

CL #00-2230

Abstract—The Tropospheric Emission Spectrometer (TES) will provide the first three-dimensional (latitude/longitude and altitude) measurements of tropospheric ozone and related species. The scientific objectives of the TES project are discussed, and an overview of the experiment and mission plan are presented. An overview of the design of the ground system is provided as context to a description of how some of the unique challenges posed by the development of the TES ground system were addressed. The solutions described include: concurrent engineering of flight and ground systems, use of CASE tools in software development, use of workstations clusters to meet computational requirements, and the development of a project-specific Framework.

TABLE OF CONTENTS

1. INTRODUCTION
2. SCIENTIFIC GOALS
3. EXPERIMENT DESCRIPTION
4. GROUND SYSTEM DESIGN
5. ASPECTS OF GROUND SYSTEM DEVELOPMENT
6. CONCLUSIONS
7. ACKNOWLEDGEMENTS

1. INTRODUCTION

The Tropospheric Emission Spectrometer (TES) is an infrared Fourier transform spectrometer slated to fly on NASA's Earth Observing System (EOS) Aura spacecraft in June 2003. The purpose of the mission is to measure the global distribution of ozone and its chemical precursors in the Earth's lower atmosphere.

TES will provide the first three-dimensional (latitude/longitude and altitude) measurements of tropospheric ozone and related species. The data will be used to calibrate current models of tropospheric chemistry, with the ultimate objective of improving those models to where they can predict future changes in the chemistry of the Earth's lower atmosphere. The measurements provided by TES will also enable researchers to search for trace gases

which may be indicative of unforeseen atmospheric processes.

Development of the TES ground system presents a number of significant technical and managerial challenges, including: a long development and operational lifetime of over 10 years; the need to develop new processing algorithms, and maintain them during the mission; significant processing resource requirements in a highly cost-constrained environment; and the need to reduce development costs while adjusting to changing budgets and schedules. Our approach to solving these problems, involving concurrent engineering of flight and ground systems, use of Computer-Assisted Software Engineering (CASE) tools in software development, use of workstations clusters to meet computational requirements, and the development of a project-specific Framework are discussed. Additional detail on many of the subjects introduced here are discussed in greater detail in [1-10].

We begin with a discussion of the scientific objectives motivating the TES Experiment, followed by a description of the experiment and the design of the ground system supporting it. Several unique aspects of the ground system development are then described.

2. SCIENTIFIC GOALS

One of the key questions in atmospheric science is "what factors control the concentration and distribution of tropospheric ozone (O₃)"? Tropospheric ozone is important for three reasons:

- 1) It is the principal component of photochemical smog. Ozone near the surface (in the so-called "boundary layer") is toxic to humans, plants and animals.
- 2) In the free troposphere (roughly 2-10 km above the surface), ozone reacts with water vapor in the presence of sunlight to form the hydroxyl radical (OH). OH, in turn, is the primary cleansing agent of the atmosphere, removing

¹ 0-7803-6599-2/01/\$10.00 © 2001 IEEE

carbon monoxide (also a toxic chemical almost totally generated by industrial activity) and other harmful chemicals such as the new hydrogenated fluorocarbons (used as CFC substitutes) from the atmosphere. In the presence of nitrogen oxides, OH is also recycled back to O₃, thus sustaining its concentration.

3) In the upper troposphere (just below the boundary with the stratosphere), ozone is a significant greenhouse gas.

Many of the chemicals involved in the formation and destruction of tropospheric ozone are quite short-lived (seconds to a few months). Their vertical and horizontal distributions in the troposphere are therefore very non-uniform and difficult to monitor from ground stations or the occasional balloon or aircraft campaign. Furthermore, there are only a limited number of such ground stations in the world, preponderantly in the northern hemisphere and all, of course, on land. The three-quarters of the Earth's surface that is ocean is essentially unmeasured.

Thus there is a clear-cut role for space-based observations of the lower atmosphere - it is the cheapest and most effective way of getting a global picture of what may loosely be termed "atmospheric pollution". Note that the problem of tropospheric ozone is quite different from the better-known stratospheric ozone problem. Tropospheric ozone appears to be *increasing* on a global scale whereas, of course, stratospheric ozone is *decreasing* with a concomitant increase in solar ultraviolet radiation reaching the surface, increasing the risks of skin cancer. Further note that a) there is roughly 10 times the amount of ozone in the stratosphere as in the troposphere and b) gas exchange between the two is slow and sporadic (indeed, the mechanisms for exchange are only poorly understood).

However, it must be emphasized that measurements in and of themselves are not the answer. The TES experiment will generate vertical concentration profiles of a significant number of the species involved in the complex chemical interactions that control the formation and destruction of tropospheric ozone. Only by utilizing these profiles in chemical-dynamical models of the atmosphere can the chemistry be said to be *understood*. There are two basic approaches to this. The more sophisticated uses a technique called *data assimilation* in which the model is actually driven by the measurements (this is how modern numerical weather forecasting is done).

Unfortunately, the field of tropospheric chemistry is insufficiently advanced to use this method at present (it is certainly planned). Instead, the current approach is to make the best inventory possible of sources and sinks of the various chemicals (on a regional, rather than local, scale) and to explore their transport and interactions using real weather patterns (obviously, after the fact!). The models are compared to the measurements and the initial conditions adjusted until the model reproduces (as well as possible) the

measurements. This method could thus be said to use measurements to *calibrate* the models. In either approach, the ultimate goal is to enable an accurate predictive capability for these models (i.e. a "chemistry" forecast).

3. EXPERIMENT DESCRIPTION

The TES experiment has two parts - the TES instrument itself and a ground data system. The space segment is an imaging infrared Fourier Transform Spectrometer (FTS). The instrument has both nadir and limb-viewing capability and covers the spectral range 650 - 3050 cm⁻¹ at either 0.0592 cm⁻¹ or 0.0145 cm⁻¹ spectral sampling distance². TES will fly on the EOS Aura platform in June 2003. (see [11] for further details on the spacecraft)

TES has 4 co-aligned detector arrays of 1x16 elements (pixels), each array optimized for a different spectral region. Each pixel Instantaneous Field-of-View (IFOV) is 0.075 mrad high by 0.75 mrad wide. At the limb, this corresponds to about 2.3 km altitude by 23 km parallel to the horizon. In the nadir, the footprint corresponds to 0.5 x 5 km. Each of the detector arrays is equipped with a filter wheel containing filters 200 - 300 cm⁻¹ wide both to reduce instrumental background noise and to permit interferogram sampling at relatively coarse intervals in order to reduce the data rate.

Table 1. Definition of Standard Data Products

Level 1A	Raw instrument data in reconstructed interferogram format, with instrument state data and geolocation data appended.
Level 1B	Calibrated spectra at full spatial and spectral resolution.
Level 2	Vertical temperature and species abundance profiles.
Level 3	Global maps of Level 2 data. One set of maps is created for every four-day global survey cycle.

TES has two basic science operating modes: Global Surveys and Special Research Observations. For Global Surveys, continuous sequences of a space view and a blackbody view calibration pair, two nadir views and 3 limb views are acquired. Calibrations and nadir views require 4 seconds each, limb views 16 seconds. Adding in the times needed for accelerating and decelerating the moving element of the FTS, each sequence requires 81.2 seconds to accomplish. 73 sequences are acquired on each orbit, triggered by passage of the orbital southern apex, and an entire survey requires 16 orbits (just over 1 day). Each survey is preceded and followed by 2 orbits of pure space and blackbody views for calibration purposes. The Aura orbit has a 16-day repeat

² The term "spectral sampling distance" is preferred over the more usual "spectral resolution" because it more accurately reflects the character of the FTS output. Furthermore, spectral resolution depends on whatever apodization may be purposefully or inadvertently applied to the data.

period so Global Surveys are made on a “1-day-on, 1-day-off” cycle.

Table 2. TES Standard Data Products

Product Name	Product Source	
	Nadir	Limb
Level 1A Interferogram	✓	✓
Level 1B Calibrated Spectra	✓	✓
Level 2 Atmospheric Temperature Profile	✓	✓
L2 Surface Temperature	✓	
L2 Land Surface Emissivity	✓	
L2 Ozone VMR Profile	✓	✓
L2 Water Vapor VMR Profile	✓	✓
L2 Carbon Monoxide VMR	✓	✓
L2 Methane VMR Profile	✓	✓
L2 Nitric Acid VMR Profile		✓
L2 Nitrogen Dioxide VMR		✓
L2 Nitric Acid VMR Profile		✓
Level 3 Gridded Product	✓	✓

Triggering from the southern apex ensures that the same locations are observed repeatedly for the lifetime of the mission. Global Surveys are the source of TES Standard Products (see Tables 1 and 2) which will be processed by the Science Investigator-led Processing System (SIPS) and archived at the NASA Langley Distributed Active Archive Center (DAAC) and are mandated by our Level 1 Requirements (see [12] for further details).

Special Research Observations fall into two general categories. The first category is targeted nadir observations of specific locations such as volcanoes or biomass burning. Such observations are made for as long as the target is within $\pm 45^\circ$ of the nadir direction (up to 210 seconds). The second category is to make transect observations: up to about 800 km long down-looking and essentially indefinitely at the limb. In every case, such observations are accompanied by appropriate calibration sequences. In general, Special Research Observations are made during the gaps in the Global Surveys. Data from these observations are processed at the Science Computing Facility (SCF). The resulting special products (see Table 3) may be archived either locally or at the Langley DAAC.

Table 3. TES Special Data Products

Compound Group	Measured Species

H _x O _y	H ₂ O ₂ , HDO
C-compounds	C ₂ H ₆ , C ₂ H ₂ , HCOOH, CH ₃ OH, PAN, CH ₃ C(O)CH ₃ , C ₂ H ₄
N-compounds	HO ₂ NO ₂ , NH ₃ , HCN, N ₂ O ³ , N ₂ O ₅
Halogen compounds	HCl ⁴ , ClONO ₂ , CCL ₄ , CCl ₃ F, CCl ₂ F ₂ , CHCl ₂ F, CHClF ₂
S-compounds	SO ₂ , COS, H ₂ S ⁴ , SF ₆

Thus TES has three clearly separable requirements on the Ground System:

- Routine processing of the Global Survey Observations. This is primarily (but not wholly) the province of the Scientific Investigator-led Processing System (SIPS).
- Special Research Observations processing. This will be conducted primarily at the JPL Science Computing Facility (SCF).
- Mission Planning. This process uses the Instrument Support Terminal (IST) which is co-located with the SCF.

In addition, the SCF and IST are used for certain mandated functions:

- Instrument characterization & calibration
- Instrument health monitoring
- Standard product quality assurance
- Support of calibration and validation campaigns & intercomparisons
- Retrieval testing (including investigation of residuals)
- Algorithm & Code upgrades

³ The vertical distribution of N₂O is known, measurement of this species is included as a control.

⁴ Volcanic plume densities only.

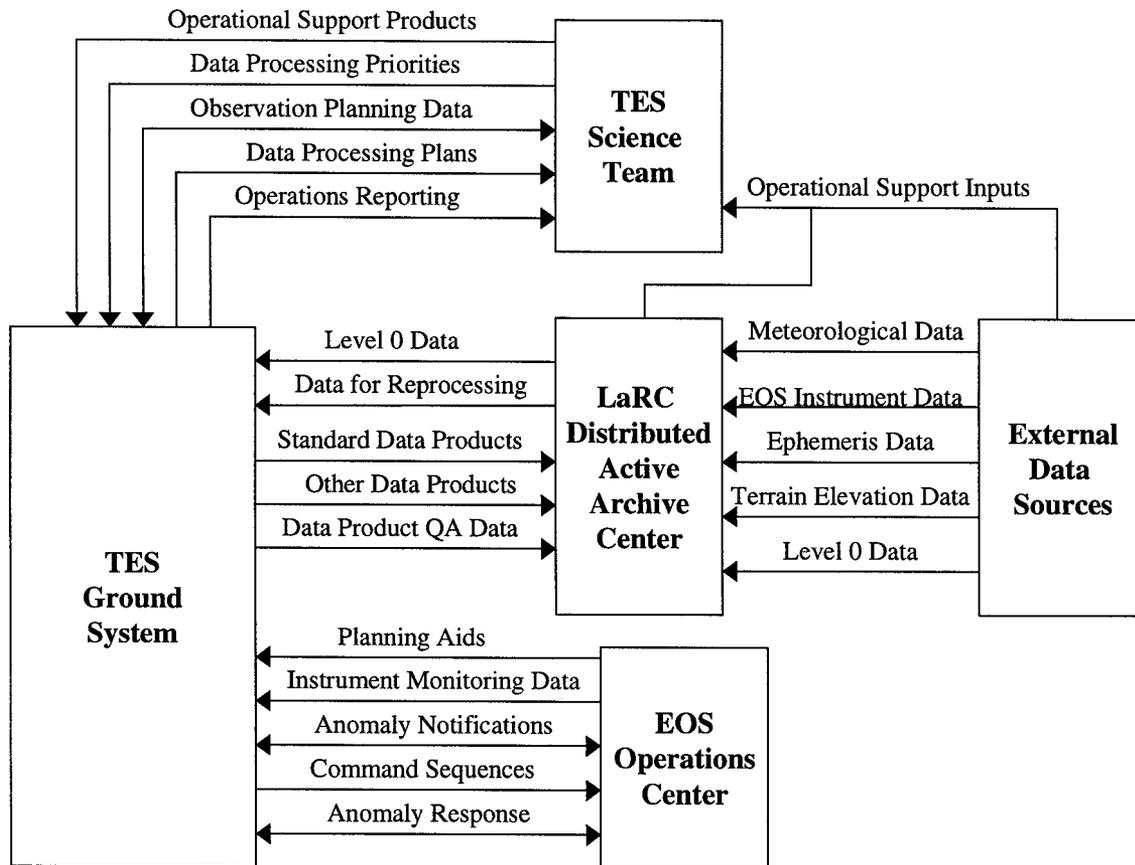


Figure 1. TES Ground System Interfaces

4. GROUND SYSTEM DESIGN

External Interfaces

The external interfaces for the TES ground system are shown in Figure 1. The activities of the ground system are directed by the TES science team, which sets priorities for mission operations and data processing, and reviews and approves operational plans. The ground system also provides the science team a means of planning global survey and special observations.

The principle interface for data processing is with the Langley DAAC. The DAAC serves as a focal point for the gathering of data sets from external sources and for the dissemination of TES data products.

The ground system interfaces with the EOS Operations Center (EOC), which is responsible for spacecraft operations. The EOC merges command loads specified by the TES mission operations team into master command loads which are uplinked to the spacecraft daily. The EOC provides 24x7 monitoring of the spacecraft and instruments, and access to downlinked instrument housekeeping data. The TES science team is ultimately responsible for the operation of TES, and uses the facilities of the ground system, via the EOC, to accomplish this.

Facilities

The hardware and software comprising the TES ground system are housed in two facilities: the Science Computing Facility (SCF), and the TES Production Facility (TPF). The SCF is collocated with the science team at JPL, while the TPF is collocated with the SIPS development team at Raytheon ITSS' Pasadena facility.

Science Data Processing System Design

The science data processing system (SDPS) is a distributed software system operating in the SIPS and SCF environment. Its purpose is to transform instrument data into standard and special data products. The SDPS comprises algorithmic processing, operational support product processing, and post-processing data quality assurance components.

The conceptual design of the algorithmic processing segment of the SDPS is shown in Figure 2. Each box labeled "PGE *n*" represents a product generation executable (PGE). A PGE is the smallest unit of processing that is independently planned and scheduled within the production system. A PGE may have subunits of execution that include staging, processing and destaging steps. The processing flow in the SCF (not shown) is more complex, but less

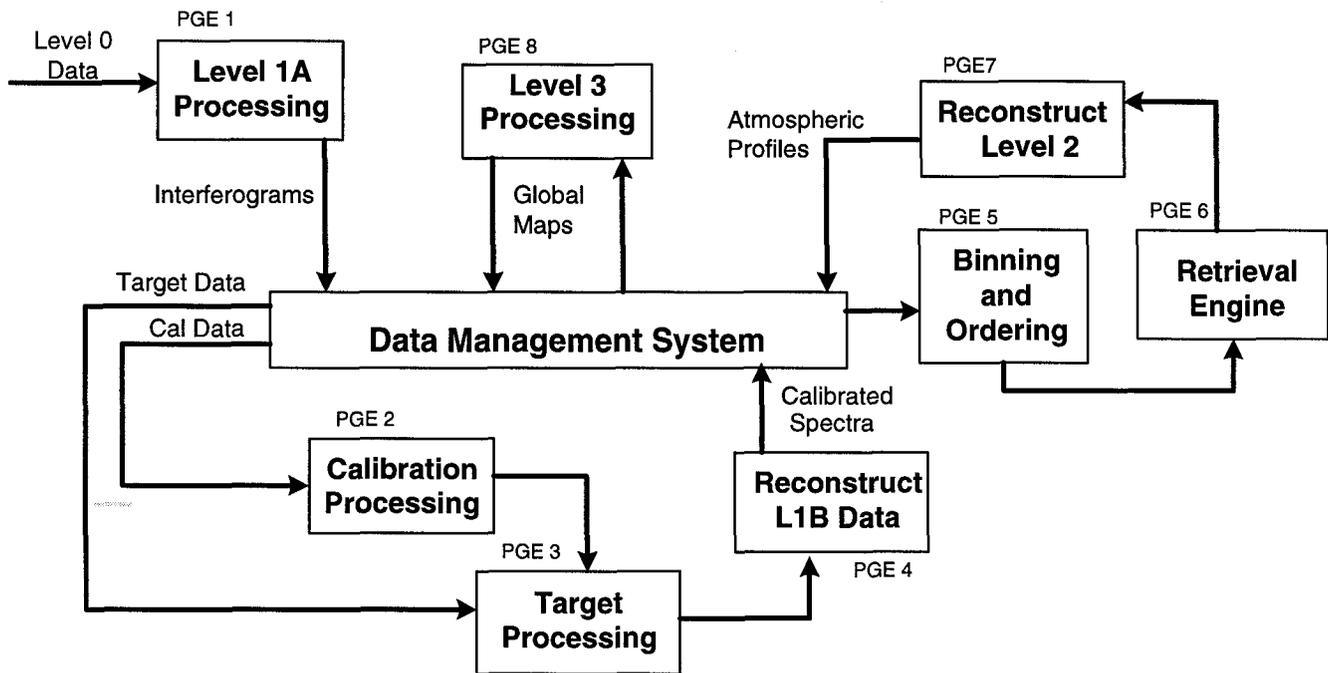


Figure 2. Science Data Processing System: Algorithmic Processing

formalized. It is expected to comprise a few dozen applications at launch.

5. ASPECTS OF GROUND SYSTEM DEVELOPMENT

Concurrent Engineering of Flight and Ground Systems

All too often, flight projects delay work on ground systems until the late stages of development. Although the early emphasis on the development of flight hardware is understandable to some degree, there is a cost of opportunity associated with doing so. Ground processes which could have been engineered for greater efficiency had they received attention when there was still flexibility in hardware design can become cumbersome to implement later in the development life cycle. The additional cost and risk of implementation may lead project managers to opt for less capable systems, relying more on manual labor or simply accepting less flexibility in mission execution.

To avoid such pitfalls, representatives of the data processing and mission operations development teams became involved in the instrument and flight software developments from very early on. The chief foci of this concurrent engineering effort were design of the data stream generated by the instrument, and development of the command interface to the instrument.

As an EOS instrument, TES was required to comply with certain telemetry interface requirements, most notably the use of the Consultative Committee on Space Data Systems (CCSDS) version 0 telemetry packets. The spacecraft interface specification further constrained the way

instruments implemented the CCSDS standards. However, within these constraints there were a number of design decisions that could be made to the benefit of the ground system team (and generally at little or no cost to the instrument), provided the proper inputs to the design process were supplied.

Data from TES' 64 detectors is generated at a rate (4.5 Mbps average, 6.2 Mbps peak) that precludes the involvement of the flight computer in packetizing the data. A special-purpose interface board, called the spacecraft interface board (SCIF) was designed to handle the incoming data, packetize it, and place it on the spacecraft's high rate interface. Data processing staff were involved in the design of the packets to ensure that ground processing of the packets could be accomplished using a set of unambiguous rules that did not require the use of information outside the packet stream. This not only reduced the cost of writing the packet processing software, it also removed an external dependency from the ground system and decreased the operational complexity.

Ground software developers worked with the science team to design the content of the science data packets to ensure that all information required to process the data were included in the high rate downlink. This entailed the addition of a fixed length "state data" region at the front of the instrument data field containing a subset of the engineering data (primarily temperature data), and the addition of two new packet types to provide information on the performance of the interferometer and pointing control subsystems.

The embedded engineering data duplicated information available in the low rate housekeeping telemetry stream, however, its inclusion in the science data enabled the data processing function to be decoupled from the instrument monitoring function. It also ensured that processing of instrument data could be performed even if the low rate data were unavailable.

Included in the state data were a set of data indices that uniquely identify a given data set. These indices are the run, sequence and scan numbers. A run is a logical unit of observation, such as a day's global survey observations, or a target of opportunity. Run numbers increment monotonically from the start of the mission. They are maintained by the flight software, based on ground-supplied information. The sequence counter identifies the number of times the basic observational sequence (such as an 81.2-second global survey sequence) has repeated in the current run. The scan counter identifies scans within a sequence. The sequence and scan numbers are entirely under flight software control.

These indices are valuable in managing data in the ground. The use of integers greatly simplifies the software required to index, search, and sort information about the TES dataset. Use of an integer handle also simplifies the task of associating data that belong to the same observation.

By using a simple increment function in the flight software, a clear and reliable relationship between observations and data may be assured. Had the indexing been done purely on the ground, the task would have been far more complex, having to take into account interruptions in delivery of data and lapses in the time ordering of the data. It would also have necessitated a laborious reading of the data, with hand-prepared data on the start and stop times of various observations. As implemented, the data arrive at the ground system with unambiguous and meaningful demarcations between related sets of data packets.

The packets containing interferometer and pointing control subsystem performance data do not duplicate data otherwise available in the housekeeping telemetry. Because it is limited to 256 bits per four-second interval, the housekeeping telemetry channel is extremely limited in the amount of information it can handle. Science processing of the data require information on the control systems that cannot be accommodated via the housekeeping channel. Working with the science team and instrument team, the ground data processing team designed a data collection and transmission scheme that met the science team's input requirements for science processing, and fit within the constraints of instrument hardware and software limitations.

Another major area of concurrent flight and ground systems engineering was the design of the command interface. The command interface encompasses the definition of the instrument commands themselves, sequencing schemes, uplink and downlink strategies, and contingency planning.

As with the housekeeping telemetry interface, the command uplink interface is limited to a relatively small bandwidth, 2,000 bps, and is available for roughly 16 ± 5 minutes each orbit. As a pointed instrument, with several other mechanical systems under control of operational sequences, TES requires a large number of commands to operate. Thus, a strategy for providing commands to the instrument had to be developed that did not unduly constrain the science mission.

The solution, described in greater detail in [6], entailed the use of stored parameter tables used to drive sequences of special commands designed to use dynamically generated parameter values. A basic looping capability was designed into the command language that further reduced the volume of uplinked commands. The table-driven strategy also makes use of the repetitive nature of most observations by using pointers to repeated information in command sequences, and expanding only the data that change frequently.

Mission operations engineers worked closely with the instrument and flight software teams during the design phases to ensure that the flight system was designed for robust and efficient of operation. An important part of this collaboration was the early emphasis on developing the operational sequences for the instrument.

The mission operations staff helped to interpret the science team's mission plan into a more concrete expression that made the operational requirements on the hardware more tangible. The net result of this effort is that the operation of TES has been simplified as much as possible, reducing the cost of operations and enabling development resources to be directed towards more value-added work such as improving planning and monitoring tools.

CASE Tools

The use of Computer Assisted Software Engineering (CASE) tools is an important part of the ground system development strategy. Software tools are used to support most of the development life cycle, from requirements management to testing. Our objectives in using CASE tools are to minimize the amount of effort required to document the system, automate as much of the testing process as possible, and ensure that key information is documented only in one place from which it may be propagated in an automated fashion.

Requirements management and design are done on Intel-based workstations running the Windows operating system. We use Rational Software's Requisite Pro, SoDA and Rose for requirements management, documentation, and software design, respectively.

Code development is done on Sun workstations and enterprise servers. Sun's Workshop development environment and GNU emacs serve as the primary programming tools, supported by McCabe IQ and Parasoft CodeWizard testing tools. Configuration management on

both the Sun and PC systems is done using CCC/Harvest from Platinum Technology.

The most innovative use of CASE tools in the ground system development is in the area of interface specification. The science data processing system entails the specification of several dozen file interfaces. These interfaces are supported by a project specific framework (discussed later in this paper). The specification of these interfaces has been integrated into the process of software design, code generation and documentation by adopting conventions for data modeling in the Rose tool and templates in the SoDA documentation tool.

We consider an interface to be a special case of a data object, which is supported by standard access mechanisms supplied by the framework. The framework provides tools for mapping an abstract view of the interface content into a physical file format, which may be binary, HDF or HDF-EOS.

File design is an implementation detail that is not part of the interface specification itself. Instead of specifying a bit-level description of a physical file, the framework data access interface is specified. This provides the user of the interface with all the information needed to read and write from it without being concerned with knowing which physical format the data have been mapped into. This allows us to separate the implementation of the interface from the process of specifying the information content.

Data interfaces are specified as object models in Rose. Using Rose's customization features, facilities for entering interface information such as data type, bit organization, and other information have been incorporated into the tool's user interface. Because the models are also used to generate code, the work required to implement an interface is minimized. There is also no translation step where a software designer is required to read an interface specification and create a separate design to implement it. We are able to save additional labor by using conventions for describing the interfaces in the model which enable us to generate a document describing the interface.

For a developer who is simply trying to use the interface and is not responsible for designing it, such documentation provides a more user-friendly way of understanding what the interface provides and how to use it than the object model provides. This approach works well for the majority of interfaces within the system. Standard product files that are delivered to outside users require substantially more information, and development of user documentation remains primarily a manual process.

Cluster Computing

There is today an important trend towards commodity cluster computing in scientific applications. The reason for this is simple: prices for powerful processors such as the

Intel Pentium, Compaq Alpha and the AMD Athlon have dropped so far that it is possible to buy the compute power of a traditional supercomputer for less than a tenth of the cost. Availability of the UNIX-like Linux operating system and the open source movement have enabled users to link these processors together into a compute environment that is robust and flexible enough to allow virtually anyone with the knowledge and the time to create a system tailored to their specific requirements.

Given NASA's mandate to reduce cost while delivering improved functionality, and the drive to increase the number of missions, cluster computing seems a natural choice. However, there are significant technical obstacles to the use of this technology in the TES ground system.

The computational cost of retrieving the atmospheric state and related parameters from the measurements made by the TES instrument is the driving requirement on the ground system processing hardware. If cluster computing is to be of use to TES, it must be capable of running the retrieval software without introducing a significant penalty on either the efficiency of the software or the cost of its development. Neither of these prerequisites is a trivial matter.

Parallel programming is a notoriously difficult task. Results of our parallelization efforts to date have been mixed. The parts of the retrieval algorithm that are well-suited to parallelization have shown near-perfect scalability in the limited tests performed so far. However, our success in those areas has spawned new technical challenges that we have not yet solved.

The chief source of difficulty lies in the amount of data required to perform the calculation. The retrieval process is designed to use precomputed absorption coefficients in order to model atmospheric radiance. Use of precomputed data reduces the computational burden to a fraction of that required to perform the computation from scratch, but the volume of that data creates a substantial I/O problem. If radiances are calculated for all frequencies measured by the instrument, roughly 5.1GB of coefficient data must be read in for each pixel and filter combination.

The overall I/O load can be reduced through the use of pixel averaging in nadir observations, and coefficient reuse in limb observations. However, these techniques greatly increase the requirement for memory and interprocess communication. Runs of the current prototype code have consumed up to 17GB of virtual memory. If use of the coefficient data is spread across several processors, then either each processor must read in its own copy of the data, thereby increasing the I/O load, or obtain a copy from another processor, which taxes the bandwidth of the network.

The preceding discussion should not be construed as a negation of the potential for cluster computing. However, it does highlight the difficulties that may be encountered in adopting it. The TES project will continue to pursue cluster computing aggressively, as the potential benefits are tremendous. An 8-node cluster of Sun SPARC 3 workstations was procured in late 2000 to support the first major test of the retrieval software. Lessons learned from that system will be applied to the procurement of a larger system to support the next major increment in testing in late 2001. These systems will in turn serve as a basis for selection of the system to be used in production processing in 2003 and beyond.

Project-specific Framework

With the current negative pressure on budgets within the federal government expected to continue for the foreseeable future, the TES project was very interested in ways to reduce development costs and to ensure the ability to provide new functionality, especially later in the project when budget pressures are expected to become more severe.

Frameworks have emerged in the last fifteen years as a means of increasing reuse and productivity. The term "framework" has many interpretations in the software engineering community. We base our use of the term on Rogers' definition [15]: A framework is, "a partially completed software application that is intended to be customized to completion." The scope of what we consider a framework includes reusable design and code components.

The decision to adopt a framework-based approach [7][9] was a strategic one, intended to fulfill the need for reduced cost, and a more robust, maintainable system. Along with the decision to develop a framework, the project adopted an object-oriented (OO) design approach and selected the C++ language. The framework decision was thus part of an overall strategy to leverage OO technology and modern approaches to reuse and development.

In our experience, software reuse is most easily achieved within the context of the originating group and their immediate colleagues. We would hope to be able to reuse our Framework on similar projects undertaken by our group, but as described below, the economic gains expected from the TES Framework are sufficient to justify its development regardless of reuse outside the project.

A reusable subroutine library was developed for the Atmospheric Emissions Spectrometer (AES), a project completed in 1993 as an airborne precursor to the TES instrument. The AES software system was considerably smaller and was not developed to the same level of standards and automation requirements of TES, but was nevertheless a substantial effort. The total system size was approximately 150 KLOC. A considerable portion of this code (roughly 105 KLOC) was already in existence when AES began, or was funded by other sources and not tracked

as part of the AES development. Detailed development records were only kept for the low-level processing and

Table 4. Summary of Framework Requirements

PGE Infrastructure An application skeleton for a PGE executable, including: command line parameter, environment variable, and user parameter processing; high-level input and output collection abstractions;

File I/O Support for I/O to/from all files in the system. Format support for HDF-EOS, HDF, ASCII and native binary file types. I/O support in terms of high-level data types (also framework supplied). Support for specific data product file organizations.

Metadata Support for EOS standard metadata output

Math Library Support Support for linear algebra, mathematical functions, specialized optimization and Fourier transforms, other functions.

Exception Handling Classes and conventions for handling exceptions.

utility code (a 26 KLOC portion consisting of two main programs and six utility programs). This portion also included the reusable library.

Of this 26 KLOC code development, about half was part of the library. Roughly three quarters of the entire AES development effort (a one work-year effort) went into producing the library and the two main programs.

Analysis of the final code sizes indicates that all of the library code was used at least twice, and 55% (comprising the data file support, file utilities, and log file code) was used in all programs. Our costs and schedule savings expectations for the TES Framework development (and subsequent rapid application development based on the Framework) have been extracted from this AES data. We found that if the 55% of code used in all programs had been developed from scratch for each use, the total system cost could have grown by as much as 180%. Looking only at the two main programs, we realized that reuse of the code would render cost savings on the order of 32% for that effort alone.

It is expected that the Framework will facilitate the development of test, data quality and other specialized tools for TES. An example of the kind of reuse we expect is the AES data extraction tool. This program required only 876 new lines of code, reusing nearly 7,000 lines of code and cutting the development time fifteen weeks to less than two. TES hopes to realize similar cost and schedule savings. If we succeed in encapsulating our higher level algorithms, we could realize similar cost savings through reuse of code, even for sophisticated applications with more complex algorithms than a simple extraction tool.

Based on the AES experience, we anticipate that developing reusable code from the outset will yield substantial economic benefits to TES.

The Framework subsystem consists of several major components designed to reduce or eliminate the necessity for the scientific processing algorithms to have information on the underlying operating system and environment. These components fall into roughly three major categories: operating system-processing interfaces, algorithmic implementation and instantiation components, and utilities. Additionally, the Framework will encapsulate various toolkits, 3rd-party packages, and commercial off-the-shelf software, in what are called Foundation components.

Many portions of the Framework could be used in a variety of applications. The domain-specific components tend to be focused around file formats and external (project) requirements, such as data archiving and process logging.

5. CONCLUSIONS

The TES experiment promises to significantly advance our knowledge of the state of the lower atmosphere, and to improve our understanding of the processes occurring there. The TES ground system is an essential part of this experiment. Implementation of the ground system has required innovative solutions to remain within an externally imposed cost and schedule cap, and to meet TES-unique challenges.

6. ACKNOWLEDGEMENTS

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

REFERENCES

- [1] *TES Ground System Operations Concept*, JPL D-18451, 2000.
- [2] Michael Lampel, Tuan Truong, Steven Friedman "Status of the TES Level 2 Numerical Analysis and Parallelization Effort," *Proceedings: 2001 IEEE Aerospace Conference*, Los Alamitos: IEEE Computer Society Press, 2001.
- [3] Brian Morrison, "Integrated Tools and Techniques Applied to the TES Ground Data System," *Proceedings: 2001 IEEE Aerospace Conference*, Los Alamitos: IEEE Computer Society Press, 2001.
- [4] Sirvard Acopyan, "Object-Oriented Software Development Based on a Detailed Science Algorithm Prototype System," *Proceedings: 2001 IEEE Aerospace Conference*, Los Alamitos: IEEE Computer Society Press, 2001.

[5] John Wright "Science Data Visualization Tools for the Tropospheric Emission Spectrometer Ground Data System," *Proceedings: 2001 IEEE Aerospace Conference*, Los Alamitos: IEEE Computer Society Press, 2001.

[6] Steven Tyler, Padma Varanasi "The Relationship Between Ground Data Systems and Flight Operations for the Aura Project TES Instrument," *Proceedings: 2001 IEEE Aerospace Conference*, Los Alamitos: IEEE Computer Society Press, 2001.

[7] Steve Larson, Stephen Watson, Kalyani Rengarajan "A Framework-Based Approach to Science Software Development," *Proceedings: 2000 IEEE Aerospace Conference*, Los Alamitos: IEEE Computer Society Press, 2000.

[8] Andy Griffin, Emily Greene, Mark Sherman, Mark Echeverri "TES Science Investigator-led Processing System (SIPS)," *Proceedings: 2001 IEEE Aerospace Conference*, Los Alamitos: IEEE Computer Society Press, 2001.

[9] Stephen Watson, Akbar Thobhani "Design and Implementation of the Tropospheric Emission Spectrometer Science Data Processing System Framework," *Proceedings: 2001 IEEE Aerospace Conference*, Los Alamitos: IEEE Computer Society Press, 2001.

[10] Sassaneh Poosti "Strategy Building Software for Atmospheric Retrievals," *Proceedings: 2001 IEEE Aerospace Conference*, Los Alamitos: IEEE Computer Society Press, 2001.

[11] <http://eos-chem.gsfc.nasa.gov>

[12] <http://www.earth.nasa.gov/visions/stratplan/index.html>

[13] TES Level 1B Algorithm Theoretical Basis, JPL D-16479

[14] TES Level 2 Algorithm Theoretical Basis, JPL D-16474

[15] Gregory Rogers, *Framework-based Software Development in C++*, Englewood Cliffs, NJ: Prentice-Hall, 1997.

Steve Larson is a software project manager at the Jet Propulsion Laboratory. He has been involved in the development of science data processing systems for several NASA Earth remote sensing projects. Prior to joining JPL, he worked as a research assistant in the areas of plasma physics and low-energy nuclear physics. He has an A.B. in Art from Occidental College and a M.S. in Physics from California State University, Northridge.

Dr. Reinhard Beer is a Senior Research Scientist at the Jet Propulsion Laboratory, and the Principal Investigator for the

TES Instrument. He has been involved in the development of several infrared Fourier transform spectrometers, including the Mark IV and ATMOS experiments. Dr. Beer is the author of *Remote Sensing by Fourier Transform Spectrometry*. He received a Ph.D. in physics from the University of Manchester, England.