

# Providing Modifiability and Robustness in the Science Software for Atmospheric Retrievals

Sassaneh Hojaiji Poosti  
Jet Propulsion Laboratory, California Institute of Technology  
4800 Oak Grove Dr., Pasadena, CA 91109  
(818) 354-3770  
sassaneh.h.poosti@jpl.nasa.gov

*Abstract*— This paper introduces Strategy software that is developed for performing atmospheric retrievals for the Tropospheric Emission Spectrometer (TES) data processing system. This software automates selection of initial parameters needed to perform an atmospheric retrieval for the Tropospheric Emission Spectrometer (TES) data processing system. Atmospheric retrievals consist of scientific algorithms used to ascertain the actual state of the atmosphere in terms of its temperature and chemical constituents.

Atmospheric retrieval algorithms are computationally expensive and convergence of the retrieval is sensitive to input, therefore it is important to provide the best input that will result in the best atmospheric profile. TES has complex requirements in terms of the variety of input data and observations. So it requires a sophisticated system such as Strategy Builder to automate the selection of the optimum strategy for performing the retrieval algorithm on its various types of observation. Strategy Software also increases the lifetime of the retrieval software by improving its flexibility in implementing various research approaches.

Strategy software uses strategy selection logic based on specific strategy selection rules for retrieval of various target scenes. This paper also addresses our approach in providing design solutions for specific challenges that existed in designing software for atmospheric retrieval software that retrieves the atmosphere of one target scene in a sequence of retrievals and performs error analysis for each retrieval.

In Strategy software, we have achieved our goals for providing robustness to TES atmospheric retrieval software. Due to this robustness and modifiability, retrieval software can be used for a longer time and more research can be conducted without the need of large additional costs of software changes.

## TABLE OF CONTENTS

1. INTRODUCTION
2. TES ATMOSPHERIC RETRIEVALS
3. STRATEGY SOFTWARE
4. STRATEGY DESIGN
5. CONCLUSION

## 1. INTRODUCTION

Many studies are performed around the subject of maintainability of the software. These studies are done for verifying the benefits of maintainability for the success of various software projects. Also, literature contains many claims concerning the difficulty of comprehending abstract models [1]. Also, it is known to every software engineer that the requirements specifications need to be known and nailed down before software design can be done. Also, it is known that the heart of requirements engineering is elicitation [3], and in projects that start with poor requirements, normally, an incremental development methodology is used to incrementally specify the requirements, design, implement and test the functionality. In our Atmospheric Retrieval application case also, like these types of projects, the incremental development approach was taken. However, in our case, poor requirements were not our only challenge, but also the high possibility of the changes in the specified requirements had to be taken into consideration.

The maintainability of the software can be in various aspects such as bug fixes, enhancement, adding new functionality, etc. This paper focuses on modifiability in the software systems such as atmospheric retrieval software systems that can benefit from it due to major requirement changes and requirement additions. It also discusses the approach that was taken in our case in the architectural and conceptual design phases to achieve the modifiability.

Science research software systems are developed to do research around new ideas. Usually in such systems, the science algorithm to be implemented for performing the research is not quiet developed when the software development is begun. Also, as the development progresses, the science algorithms are also being developed in parallel. In these type of systems, starting the development, software development team face a minimal set of requirements that are likely to change in the future. It is the nature of the science research software that requires flexibility and modifiability more than any other application. These features will allow scientists to examine various ideas and algorithms in order to obtain the satisfactory results. Therefore such software systems can greatly benefit from modifiable and flexible designs that can reduce the cost of software changes that are inevitable.

Tropospheric Emission Spectrometer (TES) is an atmospheric research project that uses a wide range of input data and more observation types compared to similar projects. TES has developed Strategy Builder software in its retrieval system architecture as a solution for providing the required flexibility for research. TES is one of the first NASA projects developing its science data processing software using object-oriented technology. This provides the opportunity to develop software that is more robust and maintainable. This paper describes the Strategy software that is developed for processing TES science data. The approach taken by TES for using the Strategy software, has great potential for usage in similar science software that require great deal of modifiability and flexibility. In addition, this paper provides some of the lessons learned in development of this software.

### *Tropospheric Emission Spectrometer*

TES is one of the instruments of the EOS (Earth Observing System) Aura mission that launched in July 2004. Aura is a part of the NASA's Earth Science Enterprise program. This program consists of missions that will help scientists to observe and understand the changes of natural processes and their effects on Earth. Oceans, clouds, land surface, and chemistry of the atmosphere are the major areas of study in these projects. EOS Aura is one of a series of polar-orbiting satellites in NASA's EOS, which is an effort to obtain long-term global observations of the land surfaces, the biosphere, solid earth, the atmosphere and oceans. TES will provide the global concentration map of atmospheric chemicals in the troposphere layer and its focus will be on the global distribution of tropospheric ozone and the factors that affect ozone formation [6, 7].

### *Atmospheric Retrieval*

Atmospheric retrieval refers to the scientific algorithms performed to retrieve the state of the atmosphere in terms of its temperature, constituents, etc. These algorithms include parts for converting the radiances from the atmosphere gases, received by the instruments, to quantities such as parts per volume of the gas. The resulted quantities are then available to other scientists for studies of the various characteristics of the Earth system.

### *TES science Data Processing*

TES ground data processing system at the time that is paper is prepared, consists of 3 processing levels [6, 7]. TES Level 1A software generates geolocated interferograms from Level 0 data packets from instrument data. In Level 1B, spectra of radiation emitted by the atmosphere will be determined from the interferograms generated by the Level 1A subsystem. In Level 2, atmospheric profiles and surface data are retrieved by using the measured spectra from L1B. ocess. In this retrieval, a model spectrum will be calculated by the Forward Model, which will be compared to the measured spectrum. Spectral calibration must be performed

so that the measured positions of molecular transitions match the calculated positions in the model spectrum.

TES uses the following iterative process for retrievals [2].

- 1 We guess a profile.
- 2 The "Forward Model" algorithm in Earth Limb and Nadir Operational Retrieval (ELANOR) process provides the radiance with respect to frequency.
- 3 If the radiance vector computed from the guess profile minus the real scanned radiance vector has a difference below a set threshold, then we decide to keep the guess profile as real profile.
- 4 If not, the guess profile is adjusted until the difference between the real radiance data and the computed data is small enough.
- 5 We repeat steps 2 to 4 until we find a match to the radiance data.

## 2. TES ATMOSPHERIC RETRIEVALS

The complete retrieval of the state of the atmosphere at each target scene requires sequential performances of the above retrieval where at each retrieval, a specific set of the atmospheric elements will be retrieved. Figure 1 shows the loop that carries out the sequential retrievals in TES Level 2 software. Each Retrieval in this loop is called a Retrieval Step. In this loop, a sequence of Strategy and input building, Retrieval, and Results processing take place. The series of Retrieval Steps that produce the complete profiles of the atmospheric constituents is called Target Retrieval. The number of times that this loop will be executed is the number of Retrieval Steps in the Retrieval Sequence. After each Retrieval Step the results will be analyzed, and based on those results the strategy and input for the next step will be built. I would like to emphasize on the concept that the strategy of each Retrieval Step is built dynamically and based on any relevant parameter including the results of another previously done retrieval.

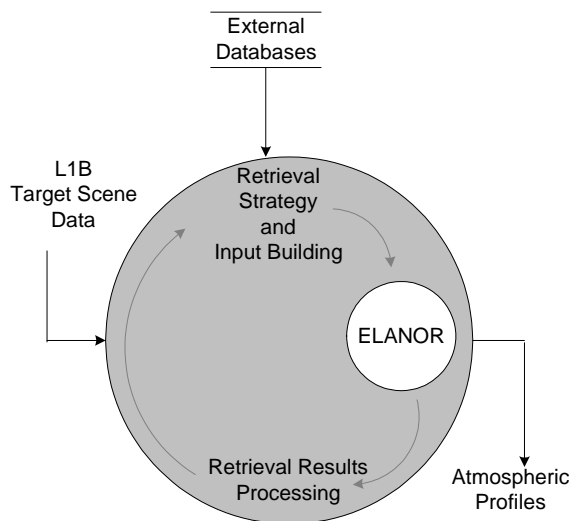


Figure 1 – Loop of Sequential Retrievals in Level 2

### 3. STRATEGY SOFTWARE

In large science software systems, although the individual science components are prototyped individually, often the science algorithm flaws are only revealed once the algorithm is integrated to rest of the system. At this time, there is chance of major changes in the implemented algorithm. Also, these types of flaws may not come to surface until several other functionality are integrated to the system. Therefore, it is greatly important to find solutions that can minimize the cost of the required modifications when the flaws are revealed.

Strategy software is a solution that is utilized in TES atmospheric retrieval software to achieve this modifiability. It also provides a high level of flexibility that TES atmospheric retrieval software requires. This solution is already implemented and the science team has utilized it for more than two years. During this period, the science algorithm and requirements have gone through many changes and more modifications are anticipated as the software is being used for processing real data from the instrument. With the changes that the software requirements have gone through during this time, the modifiable and robust design of the Strategy software has been proven effective in minimizing the cost of implementing the required changes.

In this section the objectives, concepts and design of the Strategy software is described briefly. More detailed description of this software can be found in author's paper Strategy Builder Software for Atmospheric Retrievals [2].

#### *Strategy and building input*

Strategy refers to the entity that contains the values for the set of parameters that determine the way that the atmosphere of a particular target scene should be retrieved. The specified strategy for a particular Retrieval, is used to build the input for that Retrieval.

The objective of the software that builds the strategy is 1) to select the best suitable source for obtaining the parameters' values and 2) return these data values from the selected sources. Then these values will be put together to form the Strategy. The Strategy for one retrieval program can be thought of as a set of "name=value" pairs.

#### *Target Retrieval and Retrieval Step*

The complete retrieval of the atmosphere of one target scene is called Target Retrieval. It consists of a set of consecutive retrievals. Each retrieval in this sequence is called a Retrieval Step. Because of the dependencies between various strategy parameters, data is stored in a relational database[2].

#### *Suppliers*

Strategy software uses "supplier"s to obtain data from sources that are specified in the strategy. These sources are mainly called Operational Support Products (OSPs). Each retrieval parameter may have its own supplier. Using the specified data sources, and using the guiding parameters from the strategy, suppliers process the data from the initial OSP, and provide the optimal values for all retrieval input parameter for every Retrieval Step.

Dynamically generating optimal input for each Retrieval Step is one of the most important capabilities of the TES Level 2 Retrievals. This feature is essential for the robustness of the software, since the result of each retrieval can determine or alter the strategy and input for the future steps. Using suppliers for providing the optimal retrieval input for each Retrieval Step helps providing this capability. Supplier can take the control parameters specified in the Strategy, as well as the result of the previous retrieval into account for processing and supplying data for the next step.

Suppliers are used in both Target-level and Step-level strategy building. Target-level suppliers provide target-level data and step-level suppliers provide step-level data. Figure 2 shows the general model a supplier.

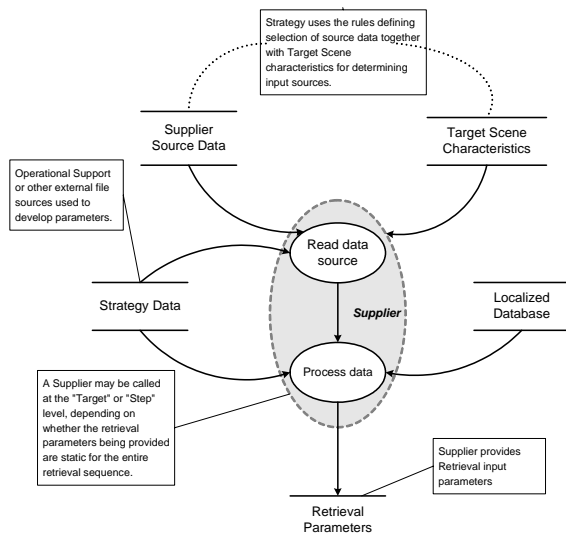


Figure 2 – General model of a supplier

### Strategy software functionality

Strategy software is responsible for performing Target Retrieval for a single target scene. Target scene is a TES footprint specified by L1A target scene geolocation information. For each one of the Retrieval Steps of the Target retrieval, the software builds a Strategy that guides the execution of the entire retrieval. This Strategy is selected and build based on the strategy selection rules provided by the scientists. Strategy selection rules are mainly based on the geolocation parameters of each target scene. Target-level Strategy contains the values for the parameters that are the same for all retrieval steps. When retrieved, some of these parameters may be updated with the retrieved values, depending on the strategy for that particular step. Strategy Builder builds the Strategy for each particular step using the target-level Strategy. It also initializes the input for that step dynamically according to the step Strategy.

Once the step-level parameters are built for one Retrieval Step, ELANOR process is invoked by the strategy software to generate calculated radiance and retrieve step-specific retrieval parameters. When the ELANOR process is over, Step Error Analysis analyzes the results and retrieval errors are reported for each Retrieval Step. Also, the retrieval results may be analyzed for potential changes to the strategy for the rest of the Retrieval Steps. Figure 3 shows the top-level Target Retrieval.

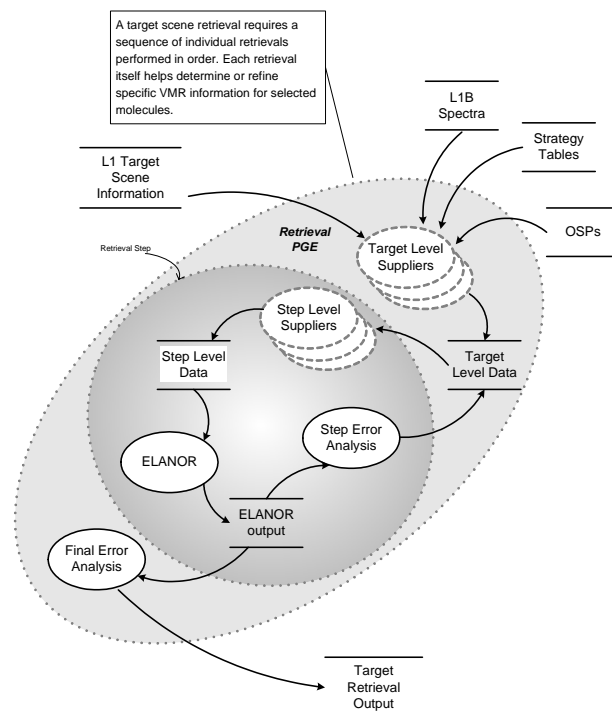


Figure 3 – Target Retrieval for one TES footprint

## 4. STRATEGY SOFTWARE DESIGN AND DEVELOPMENT

For development of Strategy software, like most of other TES ground data processing (GDS) subsystems, object-oriented paradigm is used. Strategy software has taken advantage of the information hiding, encapsulation, and polymorphism features of object-oriented design to achieve required high level of flexibility, modifiability and overall maintainability of the software. However, using object-oriented paradigm and its features alone was not enough for achieving the goal of modifiability and robustness in the atmospheric retrieval software. As mentioned earlier, in our case, like many other similar atmospheric retrieval software, the design engineer had to start designing the software with minimal known requirements. At the same time, the software designer needed to bear in mind that the requirements are very likely to be changed after the software is designed and implemented. Therefore, utilizing design techniques that help highest level of modifiability with minimal cost was important.

For coming up with such design solutions, it was necessary to understand the problem(s) that the scientists were trying to solve. Having some level of understanding of the related science was extremely helpful in foreseeing the directions in which the software may change or many need to be extended in the future. Well communication with the science team helped the understanding required for providing extensible and modifiable design solutions. In the

following sections, examples of specific design solutions will be provided. This type of analysis may also be extended at the atmospheric retrieval domain level to identify domain objects and design solutions in the domain of atmospheric retrievals [8].

### Strategy software design

Strategy software design consists of several packages. Following diagram shows a top-level class diagram of the Strategy software.

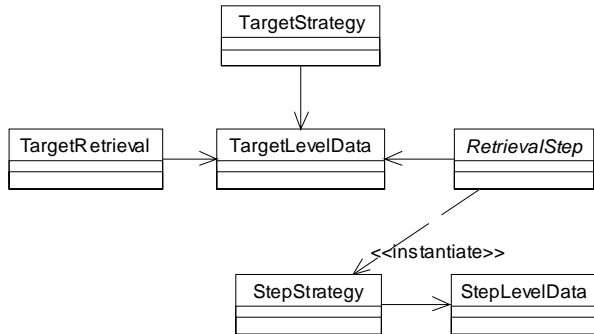


Figure 4 – Top-level class diagram of Strategy software

As mentioned earlier, in Strategy software suppliers are used both at the target-level and step-level. Figure 5 shows the relationship between the target strategy and target-level suppliers.

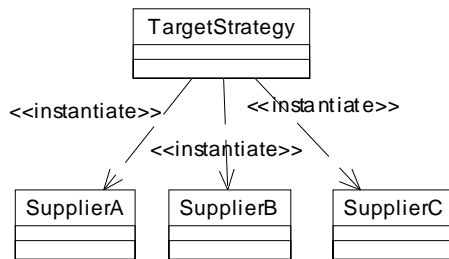


Figure 5 – Top-level class diagram of Strategy software

There are some dependencies between the data supplied by some of these suppliers and other suppliers. These dependencies determine the order of invocation of these suppliers.

In our design organization, some of the design packages contain suppliers and data classes. Data classes contain final product of the supplier and most of their objects are input to each Retrieval Step. Due to some design constraints, the Retrieval Step objects were not allowed to be dependent on the suppliers of the data. Also, the supplier objects are transient, where the data objects have longer lifetime. In fact, some of them are persistent objects.

### Step Types

In this section, a specific case is discussed about our efforts at the conceptual design phase for achieving the modifiability and robustness in the software under development. In this phase, and when initially the general and vague requirements were being specified, software developers learned from the scientists how a single retrieval works and they learned about specific input and output of a retrieval process that were known at that time. With what was explained to the software team, a single retrieval could be modeled as in figure 6.

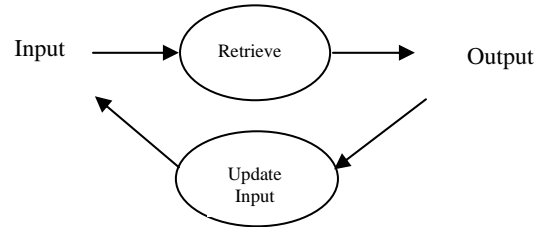


Figure 6 – Initial Retrieval Step model

Therefore, the complete retrieval would be represented as going through the loop that is shown in this figure. This seemed simple enough to design and implement. In fact, if the science algorithms and their requirements remained the same, we would probably have a simple system such as the one that is shown in this diagram. However, this model by no means can satisfy all the functionality that the software currently has.

Further communications with the scientists made it clear to the software designers that introducing the concept of a Retrieval Step can be helpful to the Strategy software development. Based on the understanding of the software team, there were several science functionality that could be implemented under the umbrella of a Retrieval Step. It was revealed that encapsulating what is currently encompasses the Retrieval Step in a class (or a set of classes) will help future potential changes to the requirements. Please note that at this time, we are talking about the changes and/or additions that are not certain. The customer was not sure whether the change would become necessary in the future. Also, these potential changes were not well known to customer. At this point, there were tradeoff issues that the software designers needed to consider. We needed to determine which solution suites the project better, a simpler design at this time that we may go through major change(s) in the future, or a more robust conceptual design that helps avoiding major design changes in the future. Since the details of the design issues, decisions and approaches are not in the scope of this paper further discussions on this subject will be deferred to another paper. Here a simplified model of the solution is represented in figure 7.

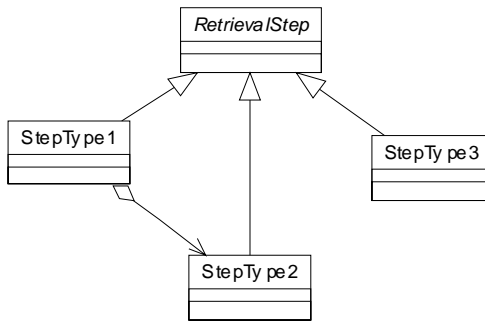


Figure 7 – Retrieval Step class diagram

In our case, generalization of the Retrieval Step has been beneficial, and several major additions and changes to the requirements were handled with significantly less effort due to generalization. The conclusion of this example is that 1) using object-oriented approach was helpful in achieving modifiability in the software design and implementation, 2) well communication with the science team and spending time for understanding the scientific problems deeper and analyzing the tradeoff has saved the cost in the long term. Also, this case, like almost all other cases, proved that domain knowledge plays an important role both in understanding and building conceptual models [1].

#### Error Analysis

Strategy software's Error Analysis component characterizes the contributions of measurement errors (statistical and systematic), "smoothing" error and representation error. This error analysis takes into account effects such as instrument noise and climatological *a priori* covariance.

At the conceptual design phase of the Strategy software, there was not much known about this component. Addition of this capability to the software was also one of the cases that were done within the minimal amount of time and effort since the concept of the Retrieval Step could be used within the error analysis context as well. The design solution for addition of this capability is one of our interesting experiences that we had in Strategy software design efforts and its result was higher than the customers' expectations. As some other components of our software, this capability is designed and implemented such that the science team can use it as a stand-alone error analysis tool as well.

## 6. CONCLUSION

TES retrieval software has developed high quality software for performing atmospheric retrievals called Strategy software. This component adds greatly to the robustness and modifiability of the TES atmospheric retrieval software. This software allows the input parameters for each Retrieval Step to be selected dynamically and the results of each Step to be analyzed for the errors, and the errors are reported along with the retrieved parameters. Strategy software modifiability and flexibility features contribute highly to

achieving the science goals of the project within the budget and schedule.

Design and development of TES Strategy software was an experience of development of large atmospheric retrieval software to be used by the scientists to experiment various algorithms and ideas for achieving accurate retrievals. In this experience, our extra efforts at the architectural and conceptual design phases have paid off in the long term. In our case, well communication between the software and science team, and spending enough time and effort for looking the scientific problems with a wider angle, and good translation of these problems to the software concepts have helped minimizing the cost of inevitable changes and additions that were made later to the requirements.

In our development, we took advantage of features in object-oriented design to provide a great amount of flexibility that was necessary for our research software. We have taken advantage of design patterns [4] where it was helpful, and have documented all the software design in UML [5] using CASE tools.

## ACKNOWLEDGMENTS

I would like to extend my appreciation to my colleagues at Jet Propulsion Laboratory especially TES Level 2 Strategy team, G. Benenyman, R. Hoffman, J. Koffend, P. Saha, W. Smith, and H. Yun for their contributions in developing the Strategy software.

This work was performed at the Jet propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

## REFERENCES

- [1] Verelst, The Influence of the Level of Abstraction on the Evolvability of Conceptual Models of Information Systems, Proceedings: 2004 International Symposium on Empirical Software Engineering, IEEE Computer Society Press, 2004
- [2] Poosti, *Strategy Builder Software for Atmospheric Retrievals*, Proceedings: 2001 IEEE Aerospace Conference, Big Sky, MT: IEEE Computer Society Press, 2001.
- [3] Wiegers, *Software Requirements*, Second Edition, Microsoft Press, 2003
- [4] Gamma, Helm, Johnson, Vlissides, *Design Patterns, Elements of Reusable Object-Oriented Software*, Addison-Wesley, June 1999
- [5] *OMG Unified Modeling Language V 1.3 Specification*, Object Management Group, Inc., Rational Software

Corp., June 1999

- [6] *TES Algorithm Theoretical Basis Document*, Version 1.1, Jet Propulsion Laboratory Internal Document D-1674, October 1, 1999
- [7] Larson, Steve, *Overview of Tropospheric Emission Spectrometer*, Proceedings: 2001 IEEE Aerospace Conference, Big Sky, MT: IEEE Computer Society Press, 2001.
- [8] Poosti, S., *Modifiability in Scientific Research software*, California State University, Northridge, 2000