TES Level 2 Production Software Design: An Object Oriented Approach

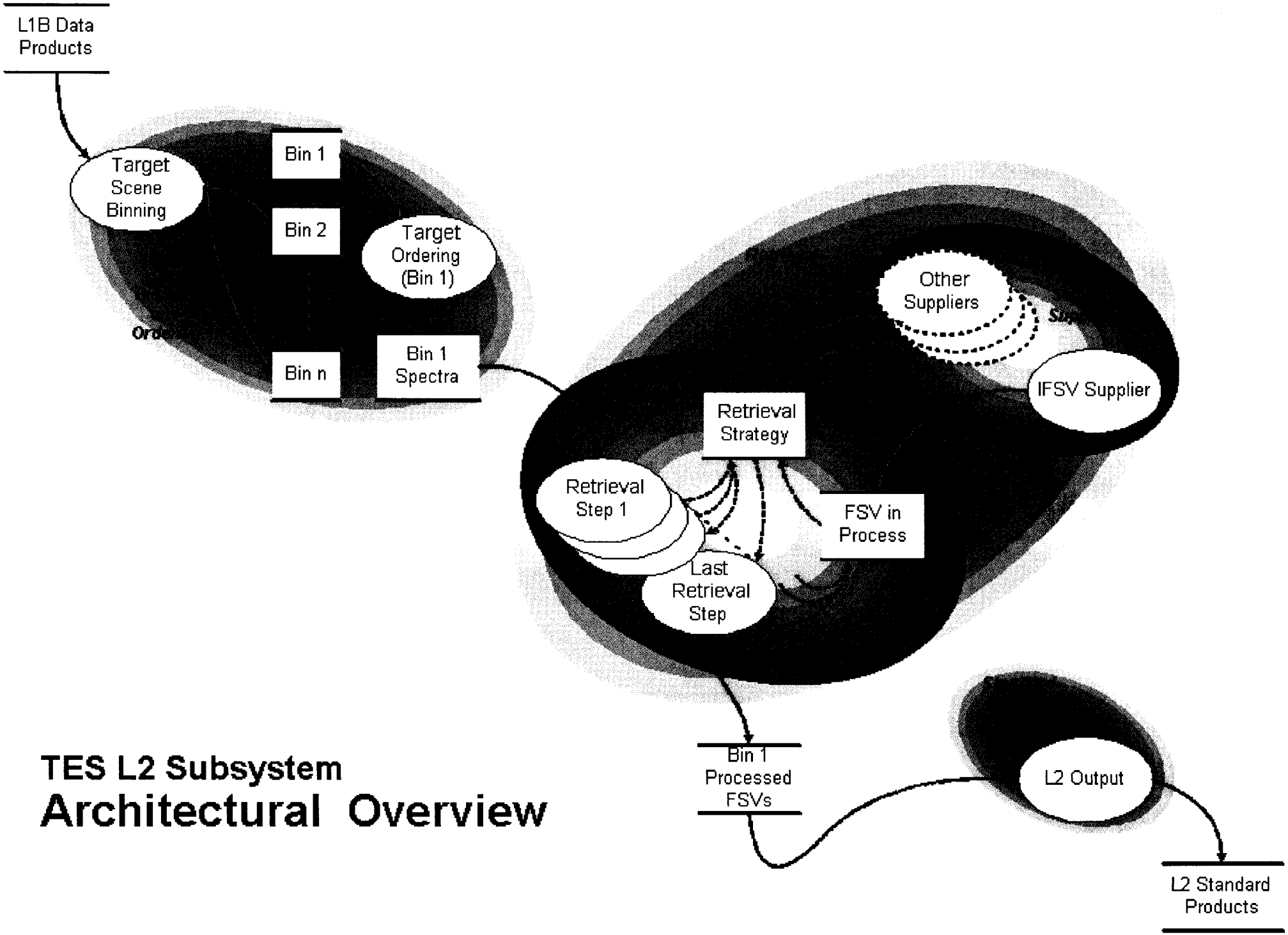
Authors: M. Lampel, S. Friedman, S. Akopyan, S. Poosti, and K. Scott

Design Goals

- Maximize Data Throughput
- Clearly Represent Atmospheric Retrieval Algorithms
- Maximize Flexibility for Updates in a Post-Launch Environment

The TES Level 2 Design Team working in coordination with the TES Science Team is currently designing the production software to encapsulate the retrieval algorithm and retrieval strategy in an object oriented style. OO approaches have been shown to be efficient for projects that are large and require a high degree maintainability and robustness. This paper will present our current status and discuss the issues driving the L2 design: data organization, algorithm, multi-processing, and performance.

Overview of TES Level 2 Subsystem



TES L2 Subsystem
Architectural Overview

An Object Oriented Approach Benefits

- Larger Implementations of Science Algorithms
- User Interfaces
- Maintainability/Reuse

Level 2 Software Development Architecture

- Target Scene Binning
- Retrieval Strategy Builder
- Retrieval
- L2 Processing Control

Impacts on Physical Architecture

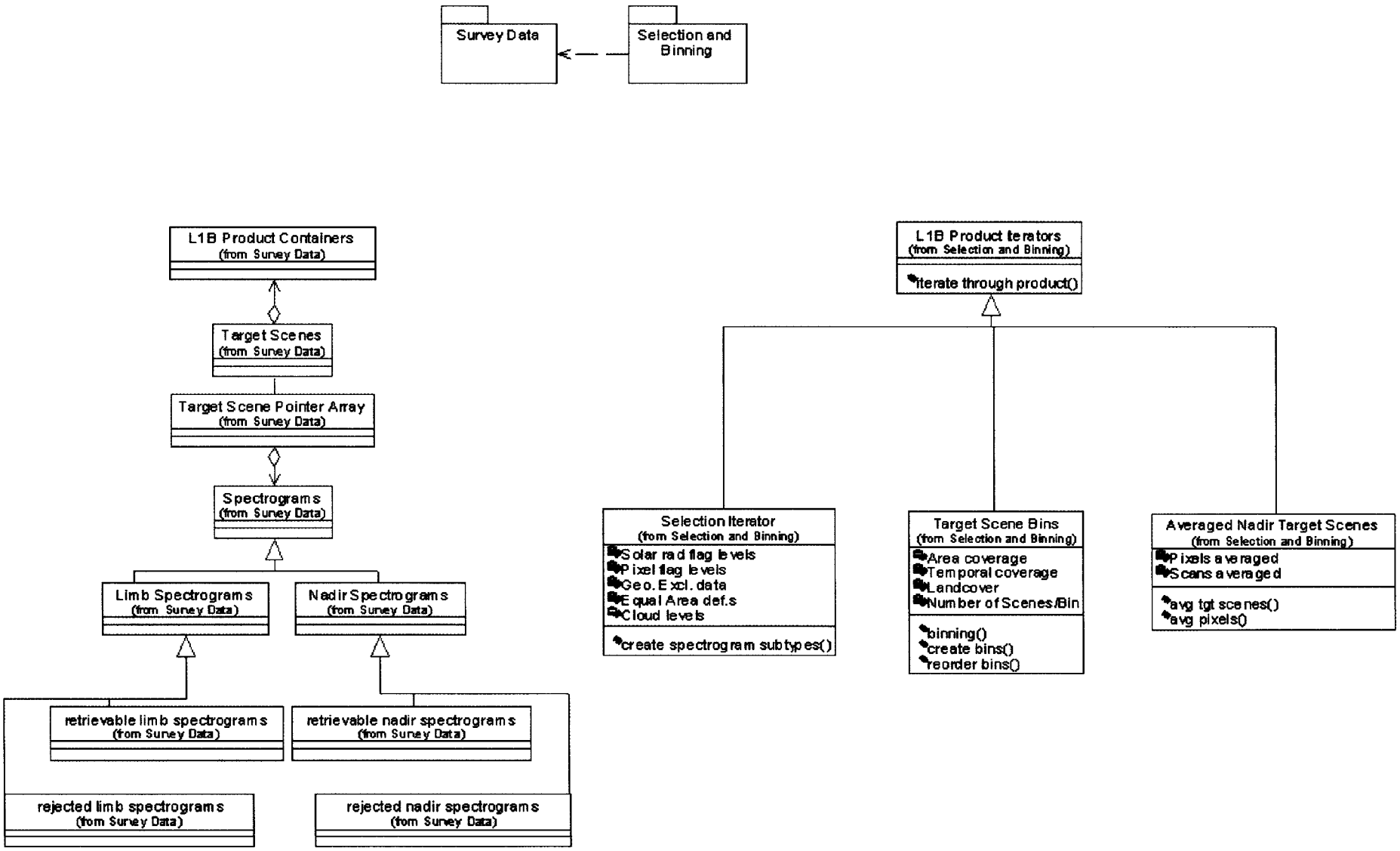
- Object Duration
- Object Memory
- Parallelization

Binning and Data Selection:

- L1B data is categorized hierarchically according to:
  (1) Relevance to L2 – e.g. retrievable or not retrievable
  (2) Viewing mode – Limb or Nadir
  (3) Self similarity criteria – e.g. geolocation, temporal proximity
- Iterator objects pass through the L1B data index
  - Act only on relevant subset of the data
  - Inherit fundamental operations
  - Specialize categorization functions
    - flexibility of algorithm updates or additions
    - reflection of a physics or analysis defined data partition

- Binning organizes data for maximum information reuse.

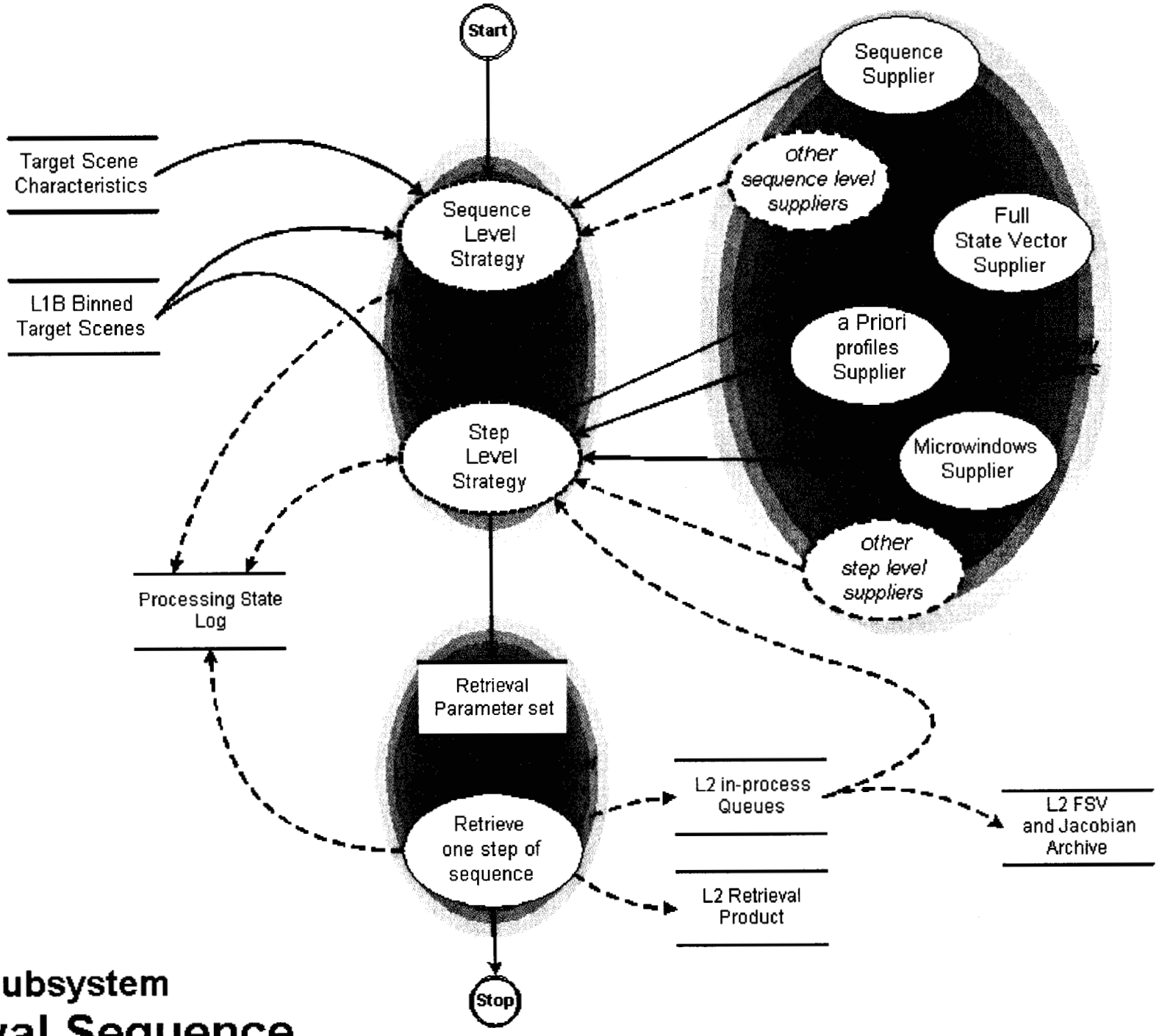
Class Structure for L1B Data and Associated Iterators


Survey Data
Selection and Binning
L1B Product Containers
(from Survey Data)
Target Scenes
(from Survey Data)
Target Scene Pointer Array
(from Survey Data)
Spectrograms
(from Survey Data)
Limb Spectrograms
(from Survey Data)
Nadir Spectrograms
(from Survey Data)
retrievable limb spectrograms
(from Survey Data)
retrievable nadir spectrograms
(from Survey Data)
rejected limb spectrograms
(from Survey Data)
rejected nadir spectrograms
(from Survey Data)
L1B Product Iterators
(from Selection and Binning)
iterate through product()
Selection Iterator
(from Selection and Binning)
Solar rad flag levels
Pixel flag levels
Geo. Excl. data
Equal Area def.s
Cloud levels
create spectrogram subtypes()
Target Scene Bins
(from Selection and Binning)
Area coverage
Temporal coverage
Landcover
Number of Scenes/Bin
binning()
create bins()
reorder bins()
Averaged Nadir Target Scenes
(from Selection and Binning)
Pixels averaged
Scans averaged
avg tgt scenes()
avg pixels()

Retrieval Strategy Builder:

- "Fills" an instruction table of retrieval inversion steps
  - rows are specified by a sequence supplier
  - columns are specified by data: Surface Parameters, Retrieval Levels, *A Priori*, Microwindows
- Suppliers implement overall Retrieval Strategy for a Target Scene
- Science Team rules encapsulated in different suppliers
- Based on relevant data and prior knowledge obtained.
- Implements data reuse for Binned Target Scenes

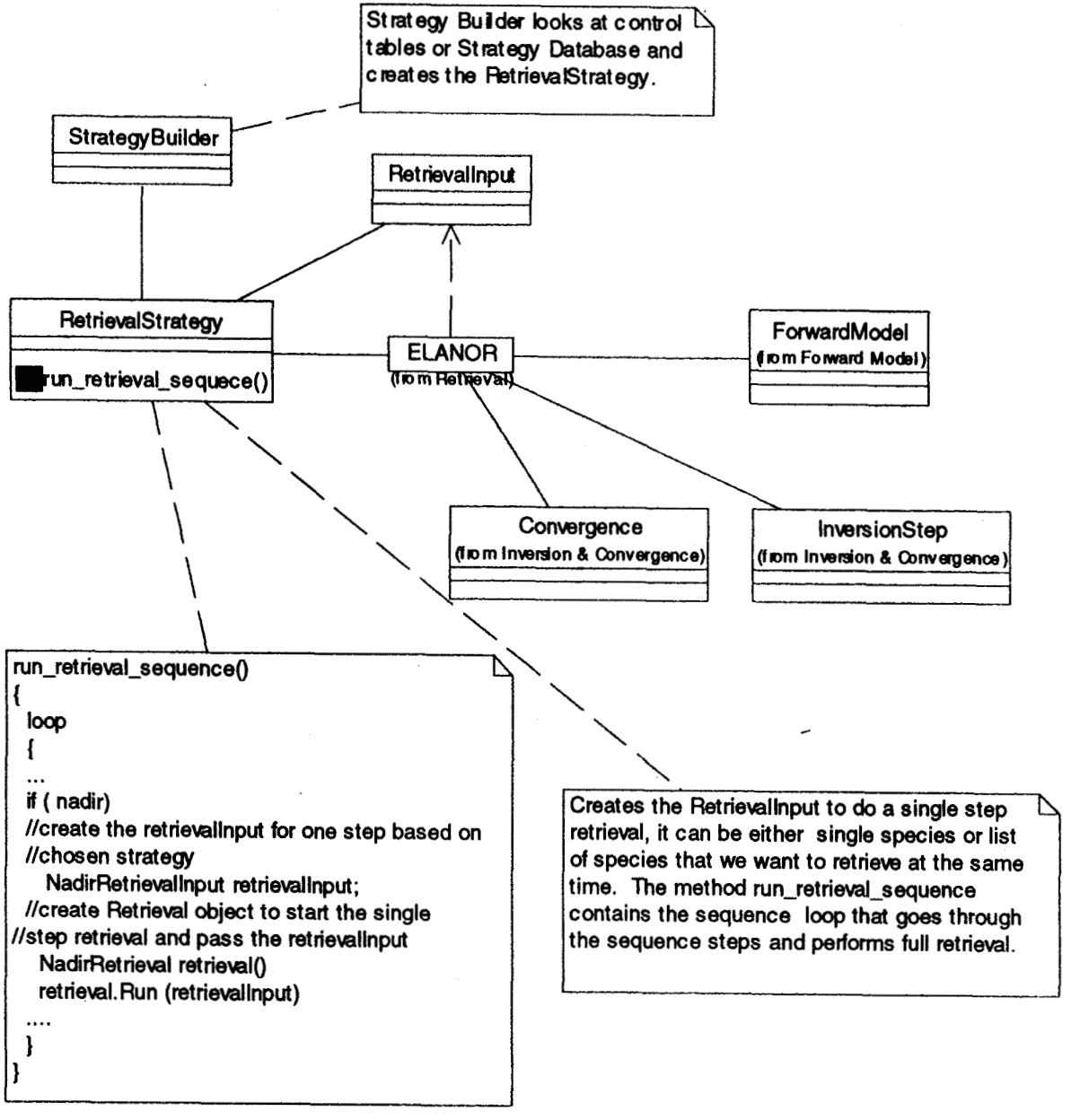
Retrieval Sequence Overview



TES L2 Subsystem

Retrieval Sequence Overview

Retrieval Object Dependencies


Strategy Builder looks at control tables or Strategy Database and creates the RetrievalStrategy.
StrategyBuilder
RetrievalInput
RetrievalStrategy
run_retrieval_sequece()
ELANOR
(from Retrieval)
ForwardModel
(from Forward Model)
Convergence
(from Inversion & Convergence)
InversionStep
(from Inversion & Convergence)
run_retrieval_sequence()
{
loop
{
...
if (nadir)
//create the retrievalInput for one step based on
//chosen strategy
NadirRetrievalInput retrievalInput;
//create Retrieval object to start the single
//step retrieval and pass the retrievalInput
NadirRetrieval retrieval()
retrieval.Run (retrievalInput)
....
}
}
Creates the RetrievalInput to do a single step retrieval, it can be either single species or list of species that we want to retrieve at the same time. The method run_retrieval_sequence contains the sequence loop that goes through the sequence steps and performs full retrieval.

Example Retrieval Strategy

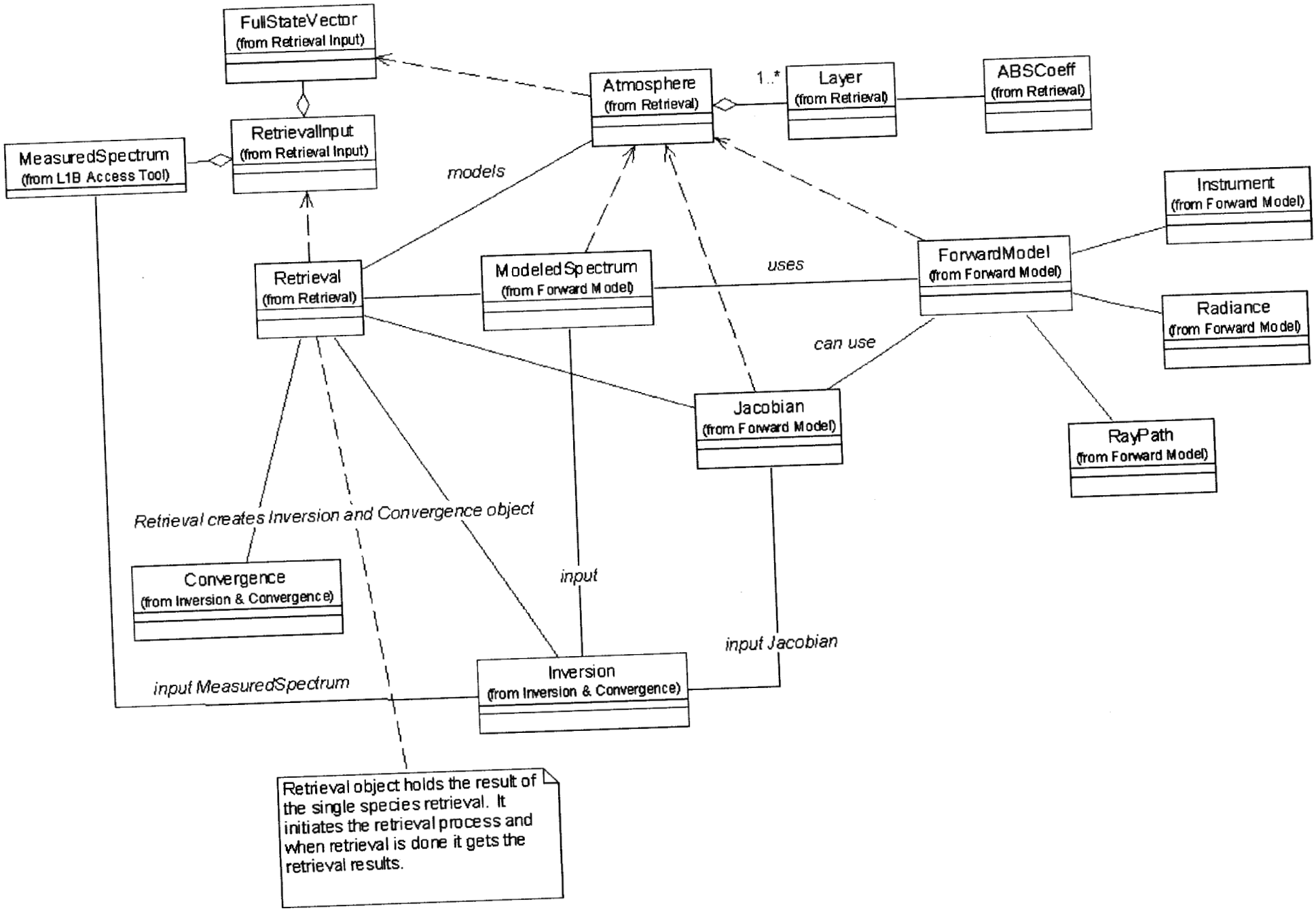
Retrieval Strategy II for Nadir retrieval parameters

Case of applicable previous retrieved data set in time/space bin

Step Index No.	Retrieval Sequence Step	Retrieval Elements	IFSV Source	Retrieval Levels	A priori Source	μwindows List	Retrieval Type	Additional FM Species	Max No. of Iterations	Convergence criteria	Inversion Method	Continuation Flag
1	T, H2O, O3, T_{surf}	T	Previous Retrieval	L1, ...		1300-1400, 1000-1100, 700-850	Normal	CO2, CFCs, E ...	2	V2	MAP/GN	T
		H2O	Previous Retrieval	L1, ...								
		O3	Previous Retrieval	L1, ...								
		Tsurf	Previous Retrieval	-								
2	Other species (NO, NO2, CO, CH4, N2O, CFC, ...)	CO	Climatology	L1, ...			Normal	T, H2O, ...	2	V2	MAP/GN	T
		CH4	Climatology	L1, ...								
		N2O	Climatology	L1, ...								
		CFC	Climatology	L1, ...								
3	Full Spectrum FM					All L1B spectra For the TGT scene	FM only	All atmospheric species	1		-	F

Retrieval:

- Encapsulates "kernel" of Forward Model/Inversion process
- Objects represent key physics or analysis information
  - Atmosphere, Atmospheric Layers
  - Rays
  - Instrument (FOV, viewing angle, etc.)
  - Jacobians
  - Modeled Spectrum
  - Retrieval Vector, etc.
- Object and class structure permits data partitioning and algorithm partitioning
  - promotes parallelization
  - numeric processing optimization

Retrieval Object Model

Main Object Model (major classes only)