ADASS XIV
ASP Conference Series, Vol. XXX, 2004
P. Shopbell, M. Britton and R. Ebert eds.

New Software for Ensemble Creation in the Spitzer-Space-Telescope Operations Database

Russ Laher and John Rector
Spitzer Science Center, MS 314-6, California Institute of Technology, Pasadena, CA 91125[1]

Abstract. Some of the computer pipelines used to process digital astronomical images from NASA's Spitzer Space Telescope require multiple input images, in order to generate high-level science and calibration products. The images are grouped into ensembles according to well-documented ensemble-creation rules by making explicit associations in the operations Informix database at the Spitzer Science Center (SSC). The advantage of this approach is that a simple database query can retrieve the required ensemble of pipeline input images. New and improved software for ensemble creation has been developed. The new software is much faster than the existing software because it uses pre-compiled database stored-procedures written in Informix SPL (SQL programming language). The new software is also more flexible because the ensemble-creation rules are now stored in and read from newly defined database tables. This table-driven approach was implemented so that ensemble rules can be inserted, updated, or deleted without modifying software.

1. Introduction

Roughly one third of the 230 downlink-data pipelines for computer processing Spitzer Space Telescope DCEs (data-collection events or raw images) rely on explicit groupings, or ensembles, of images made *a priori* in the Spitzer Science Center (SSC) operations database for *en masse* data-processing of images for creating calibration files, co-added-image mosaics, band-merged source lists, and other high-level science products (e.g., Fang *et al.*, 2003). Information about how the DCEs are grouped together as ensembles is stored in the SSC operations database so that a simple database query can retrieve the required pipeline input images. The SSC pipeline operator developed a script to perform this task, and it has been used successfully in operations since Spitzer's launch (Aug. 2003). It became apparent that the script would be more convenient if it were faster (some observing campaigns took as long 12 hours to assemble and store the ensemble relationship information). Furthermore, because the rules for ensemble creation were evolving for a long time and the script had to be modified each time the rules changed, a table-driven design for specifying the ensemble-creation rules was identified as a desideratum. This report briefly describes our next version of

[1]Author e-mail addresses, respectively: laher@ipac.caltech.edu and jar@ipac.caltech.edu.

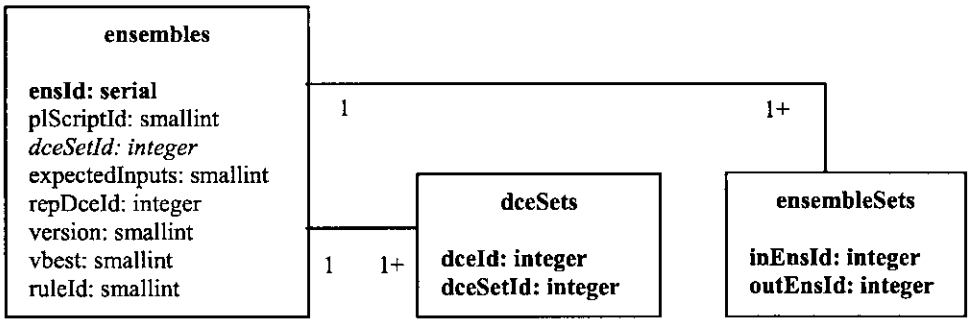

Figure 1. Schema for the ensembles, dceSets, and ensembleSets database tables.

ensemble-creation software with these improvements. The design concepts are general and can be applied to other pipeline-processing programs.

2. Database Storage of Ensembles

The SSC operations database has three tables for storing information about how ensembles of images are defined and how they are to be processed (see Figure 1). The dceSets database table associates a unique index (dceSetId) with all images (dceId) in an ensemble. The ensembles database table associates ensembles (dceSetId) with pipelines (plScriptId). The ensembleSets database table associates multiple records in the ensembles database table, which is useful for pipeline processing that requires multiple input ensembles, such as band-merging astronomical point-source lists from multiple instrument channels.

The ensembles database table is referenced by the ensembleProducts database table (not shown), in which the products of pipeline ensemble-processing are registered, thus enabling trace-back to their pipeline input images.

3. Ensemble-Creation Rules

Our design calls for storing the ensemble-creation rules in two new database tables (see Figure 2). Records in the ensRules database table specify the rules, i.e., how various attributes of DCEs are used both to query the database for specific DCEs and as discriminants for grouping the returned DCEs into one or more DCE sets. Each ensRules record has a unique index (ruleId). Records in the ensPlScripts database table specify the ensemble-processing pipelines that are to be associated with a given rule.

The table-driven feature of the design is encapsulated in the "sql" field of the ensRules database table, which stores the actual SQL query for the DCEs and their unique sets of discriminants for creating ensembles. Specifying an ensemble-creation rule requires expert knowledge of SQL and familiarity with the SSC operations database schema. However, this design allows powerful control over ensemble creation.

The "instrument" field associates the rule with the relevant Spitzer science instrument (IRAC, MIPS, or IRSX).

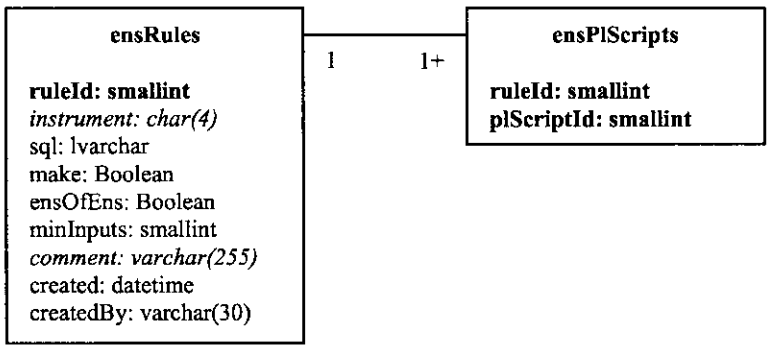

Figure 2. Schema for the ensRules and ensPlScripts database tables.

The "minInputs" field specifies the minimum number of images required in the ensemble for a given ensemble-creation rule. Ensembles not satifying this criterion will not be created and stored in the database.

The "make" field gives the capability of moving an ensemble-creation rule into abeyance without having to delete the rule from the database.

12 known discriminants of the 56 ensemble-creation rules specified thus far at the SSC are hard-wired into the new and improved software, in order to satisfy the faster ensemble-creation requirement. These rules are limited to using the known discriminants. Modifications to the software, database schema, and database stored procedures are required if additional discriminants are needed.

As part of our design, a new ruleId field has been added to the ensembles database table, in order to associate each ensemble with the rule that was applied to create it, for tracking purposes (see Figure 1). This is a distinct improvement that facilitates *post facto* analysis of how a given ensemble was created.

4. Ensemble-Creation Software

Figure 3 gives a flow chart of the software. The top-level command is implemented in perl. Although it works on one observation request at a time, it can be easily scripted to loop over all requests in a campaign. For a given request, only those ensemble-creation rules that are applicable to the associated science instrument are executed. The perl script calls our custom database stored-procedures, which use several temporary database tables to do the bulk of the work. Only the freely formulated SQL for the rules cannot be executed via database stored-procedures and therefore are executed with system calls to Informix's dbaccess command. There is a test mode that just loads the temporary database tables (not the dceSets, ensembles, and ensembleSets database tables) and outputs a very readable table, all of which provide useful diagnostics for determining whether the rule as specified is giving the expected ensembles. Finally, the software has options to 1) just give a listing of the ensemble-creation rules currently stored in the database for either a given one or all three Spitzer science instruments; and 2) print out a tutorial on the software's usage.

The software does not support multiple simultaneous users. Only one instance of the software should be executed at a time.

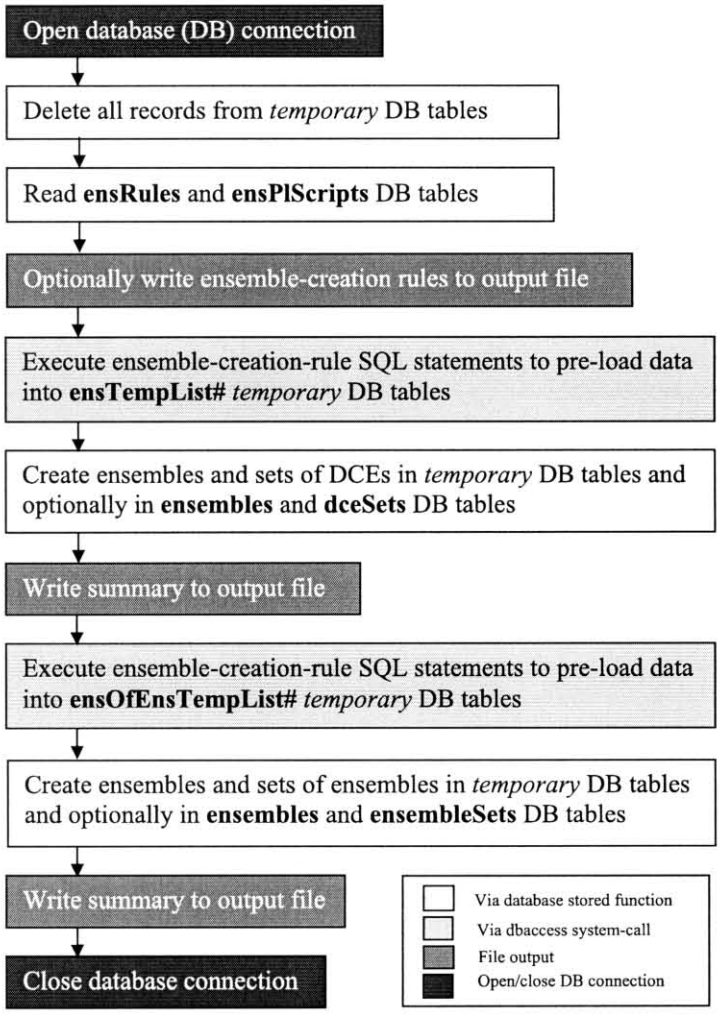

Figure 3. Flow chart for the perl script createEnsembles.pl.

Acknowledgments. This work was performed at the Spitzer Science Center as part of a mission/project managed by Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

Fang, Fan, Jing Li, Bob Narron, Clare Waterson, Iffat Khan, Wen P. Lee, John Fowler, Russ Laher, & Mehrdad Moshir 2003, in APS Conf. Ser., Vol. 295, Astronomical Data Analysis Software and Systems XII, ed. H. E. Payne, R. I. Jedrzejewski, & R. N. Hook (San Francisco: ASP), 253