

DATA MANAGEMENT USING THE EMISSIONS MODELING FRAMEWORK

Marc R. Houyoux
US EPA OAQPS (C339-02), Research Triangle Park, NC
houyoux.marc@epa.gov

Madeleine Strum
US EPA OAQPS, Research Triangle Park, NC

Rich Mason^{*}
Atmospheric Sciences Modeling Division, Air Resources Laboratory
NOAA, Research Triangle Park, NC

Alison Eyth
University of North Carolina at Chapel Hill, Carolina Environment Program, Chapel Hill, NC

ABSTRACT

The Office of Air Quality Planning and Standards (OAQPS) has created a data management tool for use in emissions modeling as part of its new Emissions Modeling Framework (EMF). This “Data Manager” supports emissions modeling data using a PostgreSQL database to store and edit data, and generate input files formatted for the Sparse Matrix Operator Kernel Emissions (SMOKE) modeling system. The Data Manager allows emissions modelers to track and access multiple versions of the same data using a version-enabled database created for the project. The Data Management part of the EMF runs on a data server, which permits multiple users to access and edit the data. The Data Manager keeps track of which users make changes, when they are made, and stores user-supplied metadata about changes to data. This paper will demonstrate this new tool, and provide examples of how OAQPS is using this tool to support air quality modeling using the 2002 National Emission Inventory.

INTRODUCTION

OAQPS is continually updating its data and tools used to support air quality modeling for regulatory analyses. Such data include emission inventories and supporting data for input to the Sparse Matrix Operator Kernel Emissions (SMOKE) modeling system, which is used to prepare emissions inventory data for input to air quality models. These data and tools are collectively called a “modeling platform”, and we are currently developing the 2002-based modeling platform. One underlying tool in that platform is the Emissions Modeling Framework (EMF)¹, and this paper describes how we are using the EMF to store, update, track, document, quality assure and access our emissions modeling data.

The Emissions Modeling Framework (EMF) is a Java-based client-server software tool with a PostgreSQL back end. Its purpose is to solve many of the long standing difficulties of emissions modeling at EPA. These difficulties stem from the complexity of emissions modeling, which requires emission modelers to piece together many disparate data sources including emission inventories, chemical speciation factors, temporal allocation factors, spatial allocation factors, growth factors, and control factors. The inventories are usually compiled from many different data sources, which can lead to inventory inconsistencies that are sometimes undesirable for modeling needs. Additionally, the inventories are usually developed separately from the speciation factors, temporal allocation factors, etc., which leads to mismatches among the datasets that emissions modelers must resolve. Without the EMF, emissions modelers must manually keep track of multiple versions of data, the modeling cases for which the datasets were used, and when/why the data were replaced by newer information. Because of the volume of data and the increasingly large number of modeling cases, such manual tracking is excessively time consuming and can be prone to mistakes, sometimes leading to problems in air quality

modeling and analysis delays. In the 2002-based platform, we seek to use the EMF to resolve many of these difficulties by using the EMF data management features, called the “Data Manager”.

Additionally, the Data Manager supports our needs for thorough documentation and traceability of data - both by those who review OAQPS regulatory work and by OAQPS data analysts. This in turn promotes transparency about decisions made during the regulatory analyses. This transparency also improves the quality of the technical work done by OAQPS and the ability for staff to quickly and accurately respond to questions about the data.

Our team in OAQPS consists of three EPA staff members who work with the emissions modeling input data on a regular basis. Much of these data are transferred to a contractor who has from 3 to 5 people who use these data to run SMOKE for several different modeling cases at any given time. In addition, the contractor also creates and modifies data at EPA’s request, and these data are then returned to EPA for review and use in existing and future modeling applications. The contractor is off-site, which necessitates that most communication happens electronically through email, which has its own set of communication challenges. A larger group of EPA staff also needs access to SMOKE output summary data. We will use the EMF to allow this team to collaborate on its data preparation, modification, quality assurance, and data summary needs. In addition, these data are also shared by team members in the Office of Research and Development (ORD) and we are currently working through some telecommunication issues to allow those team members to access the modeling platform data through the EMF.

This paper describes the capabilities of the EMF to do the following:

- Import and store SMOKE inputs in a central database
- Create, store, edit, and access multiple versions of the same dataset
- Document all that is known about datasets, including change history, problems, and resolutions
- Allow multiple users to edit the same data in a secure way
- Track steps that have been taken to quality assure the data
- Export data files for use in SMOKE

Additionally, we will describe how the OAQPS emissions modeling team will use these capabilities in support of its 2002-based modeling platform.

APPROACH

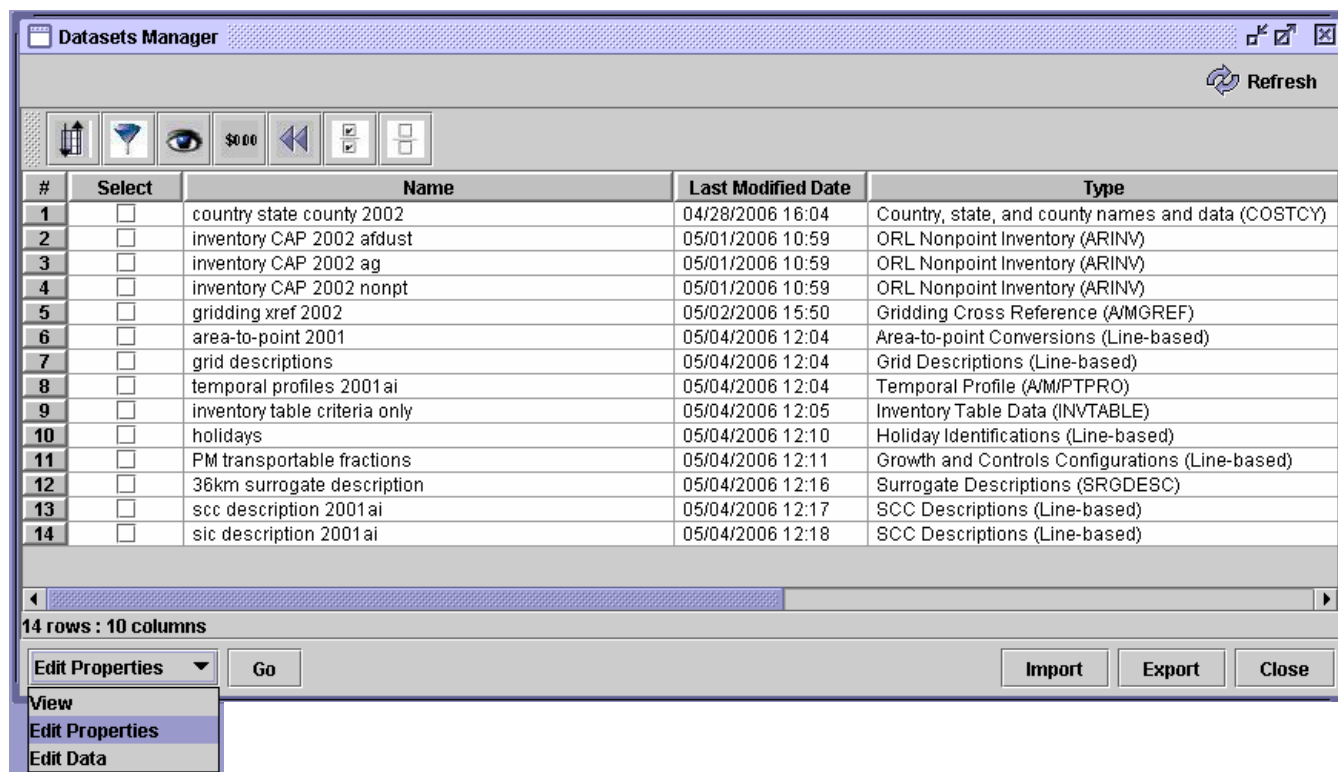
This section first describes the capabilities of the EMF and then describes how we will collaborate among EPA and contractor staff using the EMF.

Import and store SMOKE inputs in a central database

We used the Data Manager to import all of the relevant SMOKE input files we needed to use for processing the inventory using SMOKE. Once imported, these files are stored in the EMF PostgreSQL database on the EMF server. All emission modelers in OAQPS can access the same copy of the same data, which ensures that everyone is using the same information. This data sharing also prevents multiple copies of datasets from being created so that the entire team knows which copy is the “official” data. Figure 1 shows the Data Manager main window, which provides a table of all of the imported datasets and key relevant information about them. The Data Manager also supports data formats such as the National Inventory Format (NIF), which has multiple tables associated with a single dataset. These are treated by the Data Manager as a single “Dataset” even though these data are imported as several related files. The Data Manager also supports a generic comma separated variable (CSV) file, which

allows many other types of data to be imported and managed in the EMF. The toolbar buttons at the top of the window also permit sorting, filtering, and formatting of the table.

Figure 1: Datasets Manager main window



In addition to managing individual datasets, the EMF also allows users to define and maintain “Dataset Types”. A Dataset Type serves a number of purposes, including defining:

- A table structure in the EMF Database
- A format used by the EMF for importing and exporting the data to/from text versions of the data
- Keywords that are required for that Dataset Type
- Quality Assurance (QA) steps defined for that Dataset Type (both required and optional steps)

New Dataset Types can be added by the user as long as they are based on one of the four “generic” types (CSV, line-based, external, and SMOKE report). The “line-based” type is for storing complex formats that have different structures for each line of data; an example for SMOKE users is the control and projection packets file (GCNTL). The “external” type is to be able to reference and store metadata about a dataset, but that dataset is not actually stored in the EMF database; rather, it is stored on disk and referenced by its location. This type supports large data that are not necessarily desirable for inclusion in the database, such as meteorology data files or shapefiles used to create spatial surrogate data. The “SMOKE report” type is the output format from SMOKE’s Smkreport program used to create summaries for QA. While it is useful to allow users to add Dataset Types based on these four types, adding other Dataset Types requires a Java programmer. Once a user adds a Dataset Type, then all users can import data of that type, taking advantage of the predefined format, keywords, and QA steps.

Create, store, edit, and access multiple versions of the same dataset

With the Data Manager, users can create new versions of the data without making a copy of the entire dataset. Each change in a dataset is associated with a version, and only the changes from the previous version are stored. This approach provides for efficient storage of a large number of changes and versions, since the original data is not copied each time a new version is needed.

When a user wants to make a change to a dataset, (s)he selects “Edit Data” from the Datasets Manager interface and clicks “Go”. The Data Manager then shows a Dataset Versions Editor table that summarizes the available versions, as shown in Figure 2. With a new dataset, only a single entry called “Initial Version” appears in the table shown in Figure 2. To create a new version, the user clicks “New”, and the Data Manager adds a new version to that table (shown below, this version name is “Initial with VA change”). The user must also select which version to use as the basis for the new version; this feature enables a “tree” of versions to be created instead of being limited to a linear history of versions. The user-defined name appears in the “Name” column of the table and the automatic version number appears in the “Version” column. The “Base” column indicates the version number upon which that version of data is based. Once the new version has been created, the user can choose the “Edit” button to display and edit the data using the Data Editor interface shown in Figure 3.

Users also can set a version of a dataset as “final”, as indicated by the “Is Final?” column in Figure 2. After a dataset version is final, no additional changes to it can be made. This approach has been chosen to prevent changes to datasets that may have already been used. We have made the assumption that a “final” version may have been used for some purpose such as input to an air quality model, which means that any changes to that version would make it inconsistent with that use. So, while users can create another version based on a “final” version, they cannot change any version marked “final.”

Figure 2: Dataset Versions Editor table

The screenshot shows a window titled "Dataset Versions Editor: inventory CAP 2002 ag". Inside, there's a section labeled "Versions" with a table. Above the table is a dropdown menu showing "DS_00_NEI_Area_Agricultural_NH3_ag_CEP_09mar_006_ORL" and buttons for "View" and "Edit". The table has columns: "Select", "Name", "Version", "Base", "Is Final?", and "Date".

Select	Name	Version	Base	Is Final?	Date
☒	Initial Version	0	0	☒	04/28/2006
☐	Initial with VA change	1	0	☐	05/09/2006

Below the table are buttons for "New" and "Mark Final". At the bottom right is a "Close" button.

The Data Editor (shown in Figure 3) provides a number of options for editing data. Users can add rows, delete rows, and changes any values in the data. The interface supports sorting and filtering of the data to make it easier for users to find the data of interest for making changes. The interface automatically checks as the user enters data to make sure that the values entered have the correct type (e.g., integer, real, character) and length as what is expected for the Dataset Type. Additionally, the Data Editor has been designed to “page” large data for better performance. Only one page of data is loaded into the editor at a time, though the Editor can rapidly get additional pages as requested by the user using the scroll bar shown on the upper right of Figure 3.

The Data Manager provides the capability for users to document changes using the data editor, which appears as the “Revision Information” part of the window shown at the bottom of Figure 3. The user is required to enter a description in the boxes labeled “What was changed” and “Why it was changed”. The Data Manager automatically records the user name and date/time of the change. The “who”, “when”, “what” and “why” information is associated with a particular change and a specific version; therefore, users can always determine differences between versions.

Figure 3: Data Editor interface

Data Editor: inventory CAP 2002 afdust, Version: Version 2, Table: DS_00_NEI_Area_Fugitive_Dust_afdust_CAP_09mar_006_ORL

Lock expires at 13:36 05/04/2006

Sort Order:

Row Filter:

Current: 1 - 100 Filtered: 41219 of 41219

Apply

SELECT	FIPS String(6)	SCC String(10)	SIC String(4)	MACT String(6)	SRCTYPE String(2)	NAICS String(6)	POLL String(16)	ANN_EMIS Double
☐	12087	2275085000			02		PM10-PRI	2.74
☐	12087	2275085000			02		PM25-PRI	0.41
☐	12021	2275085000			02		PM25-PRI	1.23
☐	12021	2275085000			02		PM10-PRI	8.22
☐	12083	2275085000			02		PM10-PRI	4.11
☐	12083	2275085000			02		PM25-PRI	0.62
☐	12099	2275085000			02		PM10-PRI	5.48
☐	12099	2275085000			02		PM25-PRI	0.82
☐	04001	2294000000			02		PM10-PRI	1589.29
☐	04001	2294000000			02		PM25-PRI	161.49
☐	30003	2294000000			02		PM10-PRI	309.27
☐	30003	2294000000			02		PM25-PRI	34.152

Revision Information

What was changed:

Why it was changed:

References:

Add Note

Discard Save Close

As additional changes are made to a dataset, users can use the Data Manager to access all of the previous versions of the data. This is useful, for example, when a particular version was used in a modeling case, and other people want to look at the summary of that data or attempt to reproduce what has been done in the past. To access past versions of data, users can either use the Dataset Version Editor shown in Figure 3, or the “Data” tab of the “Dataset Properties View”, as shown in Figure 4. This second option allows users to view the data without accidentally editing it. If the version is final, however, users cannot edit the data without creating a new version.

Document all that is known about datasets, including change history, problems, and resolutions

The Data Manager provides an extensive metadata capability, which includes the following types of metadata. Each of these metadata types corresponds to a tab on the “Dataset Properties Editor” (when editing the metadata) or on the “Dataset Properties Viewer” (when simply viewing the information). Users can access the former from the Datasets Manager interface (Figure 1) by selecting “Edit Properties” and clicking “Go”, while users can access the latter by selecting “View” and clicking “Go”.

- Summary data (Figure 4). This metadata includes entries that record a description, project name, user who created the dataset, Dataset Type, time period, temporal resolution, sector, region, country, status, last modified date, last accessed date, creation date, intended use, and which version is the “default” version. For example in Figure 4, the time period applies for the whole year and therefore a user has set the temporal resolution to “Annual”. Also, the user has set the region to “national” and the country to “US”. The EMF has automatically recorded the last-modified date, the last-accessed date, and the creation date. These data are not maintained as version-specific by the Data Manager. A status of “imported” is currently the only available

status, but this part of the metadata will be used to indicate QA and usage statuses in future versions of the EMF.

The user can set the “intended use” setting to any phrase to indicate how the user expects the data to be used. This can be interpreted by the user to be defined based on project names, modeling case names, or to indicate “public” data. Since there not yet a capability for the Data Manager to lock other users out of a dataset, OAQPS plans to use the “intended use” capability to indicate “public” or “private” data to other users on the team.

Figure 4: Dataset Properties Editor Summary tab

The screenshot shows the 'Dataset Properties Editor' window for a dataset named 'inventory CAP 2002 afdust'. The 'Summary' tab is selected, showing various metadata fields. The 'Name' field contains 'inventory CAP 2002 afdust'. The 'Description' field contains a list of keywords: '#ORL NONPOINT', '#TYPE afdust Inventory for CAPS', '#COUNTRY US', and '#YEAR 2002'. The 'Project' field is set to '2002-based platform'. The 'Creator' is 'mhouyoux'. The 'Dataset Type' is 'ORL Nonpoint Inventory (ARINV)'. The 'Time Period Start' is '01/01/2002 00:00' and the 'Time Period End' is '12/31/2002 23:59'. The 'Temporal Resolution' is 'Annual'. The 'Sector' is 'afdust'. The 'Region' is 'National' and the 'Country' is 'US'. The 'Status' is 'Imported'. The 'Last Modified Date' is '05/01/2006 10:59' and the 'Last Accessed Date' is '04/28/2006 14:59'. The 'Creation Date' is '04/28/2006 14:59'. The 'Intended Use' is 'public' and the 'Default Version' is '0 - Initial Version'. 'Save' and 'Close' buttons are at the bottom right.

Dataset Properties Editor: inventory CAP 2002 afdust	
Summary Data Keywords Notes Revisions Logs Tables QA	
Name:	inventory CAP 2002 afdust
Description:	#ORL NONPOINT #TYPE afdust Inventory for CAPS #COUNTRY US #YEAR 2002
Project:	2002-based platform
Creator:	mhouyoux
Dataset Type:	ORL Nonpoint Inventory (ARINV)
Time Period Start:	01/01/2002 00:00
Time Period End:	12/31/2002 23:59
Temporal Resolution:	Annual
Sector:	afdust
Region:	National
Country:	US
Status:	Imported
Last Modified Date:	05/01/2006 10:59
Last Accessed Date:	04/28/2006 14:59
Creation Date:	04/28/2006 14:59
Intended Use:	public
Default Version:	0 - Initial Version
Save Close	

- Keywords (Figure 5). Users can create and use whatever keyword system they like to help organize their data. Some preset keywords are recognized by the EMF for use in exporting data, such as the “EXPORT_PREFIX” and “EXPORT_SUFFIX”, and these are shown as an example in Figure 5. When a user exports the dataset shown in this example, the file name would start with “arinv” and end with “_orl.txt”; the middle of the file name would be the Dataset name with spaces and special characters replaced by underscores. Other preset keywords allow you to define relationships between datasets: “REPLACES_DATASET” allows users to indicate if that dataset replaces another dataset and “REPLACED_BY_DATASET” allows users to indicate which dataset replaces that dataset. The keywords apply to all versions of a dataset.

Figure 5: Dataset Properties Editor Keywords tab

Select	Keyword	Default Value
☐	EXPORT_SUFFIX	_orl.txt
☐	EXPORT_PREFIX	arin

- Notes (Figure 6). Users can also add Notes to provide documentation about the datasets. Notes can be categorized into “Observation”, “Replaces dataset”, “Replaced by dataset”, “Question”, “Answer”, “Problem”, and “Resolution”. Users can also reference one note from another note, which can be used to link a problem with a resolution, or a question with an answer, for example. For example in Figure 6, user “Marc Houyoux” has added a note “What SCCs are included in this data file?” as a “Question” and a response listed as “SCCs that are included in this file are...” has been added as an “Answer”. Users assign each note to a version of a dataset, and in this example, all notes are applied to Version 0.

Figure 6: Dataset Properties Editor Notes tab

#	Select	Id	Summary	Type	Version	Creator	Date
1	☐	4	Adding an example note	Observation	0	Marc Houyoux	05/04/2006
2	☐	5	What SCCs are included in this data file?	Question	0	Marc Houyoux	05/04/2006
3	☐	6	SCCs that are included in this file are...	Answer	0	Marc Houyoux	05/04/2006

3 rows : 9 columns

- Revisions: The Revisions tab provides a summary of who created a new version and when, as well as the documentation entered about a version when the version was changed (i.e., the “What” and “Why” fields shown previously in Figure 3).
- Logs: The Logs tab provides a summary of all of the “uses” of a dataset. Currently, the only “use” of a dataset is that it has been exported; however, a future version of the EMF will allow “uses” that include running the emissions through the SMOKE system to create future-year emissions or model-ready emissions. The log includes who used the data, when, what version was used, and a description of the purpose of the export. For example, users can indicate if they were exporting the data for a specific project. In the case of exporting data, the log also indicates to what directory the data file was exported.

- **Tables:** The Tables tab lists the tables of a dataset. In most cases, only one table exists for a dataset. In some cases like NIF3 files, multiple tables would be listed on this tab. The same tables must be included for all versions of a dataset.
- **QA (Figure 7):** The QA tab provides a list of the quality assurance steps that the user has entered for the dataset, along with the status of those steps. This tab allows the user to add additional QA steps in addition to pre-defined steps associated with the Dataset Type. This metadata is specific to each version of the data and includes the description of the step, when it was performed, who performed the step, and comments made by the user about the results of the QA step. For example in Figure 7, two QA steps have been added to this dataset: “Validate SCC List” and “Create County Summary”. The first step in Figure 7 is a required step (as denoted by the check in the box under the “Required” column) and the second is an optional step. The QA feature is explained more in the section below called “Track steps that have been taken to quality assure the data” and in a companion paper².

Figure 7: Dataset Properties Editor QA tab

#	Select	Version	Name	Required	Order	Status	When	Who	Comments
1	☐	0	Validate SCC List	☒	1.0	Not Started	N/A		
2	☐	0	Create County Summary	☐	2.0	Not Started	N/A		

Allow multiple users edit the same data in a secure way

Multiple users can access the EMF simultaneously, but this feature causes challenges for ensuring that data are not inadvertently corrupted by two users attempting to modify the same dataset at the same time. To avoid this problem, the Data Manager prevents two users from attempting to edit the same dataset at the same time. To do this, the Data Manager initiates a one-hour “lock” on the dataset when editing is started. As shown in Figure 3, the lock expiration time appears at the top right of the Data Editor window. The Data Manager updates the lock start-time each time a change to the dataset is saved, which essentially allows the Data Manager to distinguish between a user who is actively editing a dataset and a user who has forgotten to close their editor window, but has stopped editing the data. Other users may still edit the dataset properties or view versions of the data that are not being edited.

The EMF has also been designed to prevent worse-case scenarios of a user starting to edit a dataset (thereby initiating a lock) and then leaving for an extended period such as a business trip. Once the lock expires an hour after the last saved edit, other users will be able to take over the editing session. All saved changes made by the previous user would be available to the other users, although any unsaved changes are not stored on the database and would therefore be lost. This implementation still requires communication among team members and would not be useful if multiple unrelated groups were working together on the same data. In future versions of the EMF, we hope to support user groups and permissions for sharing data. We also plan to implement in a future version the ability for the

administrator to override a lock at any time. These features overcome a potential collaboration challenge created by the otherwise extremely useful feature of having multiple users able to work on the same datasets.

Track steps that have been taken to quality assure the data

A component of the EMF that works in close collaboration with the Data Manager is the QA Manager. This part of the EMF is accessed through the setting of QA steps for a Dataset Type and also through the QA tab on the Dataset Properties Editor (Figure 7). A companion paper at this conference² is entitled “Quality Assurance of Emissions Inventory and Modeling Data with the Emissions Modeling Framework and EmisView” and describes this feature in more detail. We point out this capability here because it is part of the discussion below regarding how we are planning to use the EMF. The QA Manager allows users to indicate which QA steps have been completed and make notes about the progress, including setting an overall QA status such as “Not started” (the default), “Skipped”, “In Progress”, “Complete”, and “Failure”. Figure 8 provides a picture of the interface that users access to edit information about a QA step.

Figure 8: Edit QA step interface

Edit QA Step: 1 Validate SCC List - inventory CAP 2002 afdust *

Name: Validate SCC List

Version: 0 Initial Version

Program: EmisView

Arguments: scc subset

Order: 1.0

Required? ☒

Description: Compare SCCs from a summary file to the master EMF SCC list. All SCCs must be valid before this step can be marked as "Complete"

Status: Complete

User: Marc Houyoux

Date: 05/09/2006 10:50

Configuration: repconfig_scclist.txt

Comments: Compared SCCs from summary file to master EMF SCC list and they matched

Buttons: Save Close

OAPQS plans for using the EMF to support its 2002 modeling platform

OAPQS plans to use the Data Manager and QA Manager in support of its 2002 modeling platform to store the SMOKE input inventories and ancillary data, collectively called the SMOKE input data. Using the EMF, any team member could import the SMOKE input data into the system. Initially, EPA staff will load the SMOKE input data into the system as well as inventory summary data (such as state totals of the inventory). As we become accustomed to using this tool, we anticipate that both EPA and the contractor will make changes to the data using the Data Manager. Since old versions are maintained and the EMF facilitates documentation on the changes that are made, we will be much better able to check each others work to prevent erroneous data changes. Also, if an error is made, it will be clear what has happened. To use the data in SMOKE, the user must export data from the EMF first – though we are working towards a version of the EMF that runs SMOKE and gets the data from the database automatically.

EPA staff will also define quality assurance steps for each of the Dataset Types being used by the 2002-based platform, and they will load these into the EMF. Prior to using the data, the contractor and EPA staff will be responsible for performing these QA steps and recording the results of that QA in the EMF. We plan to require all QA steps to be completed prior to a dataset being labeled as “final”. Once the QA has been performed and the datasets are final, EPA staff will set the “Default version” to set the version to be exported. Then, EPA staff will provide a list to the contractor of all datasets to be used for a particular modeling case. The contractor will export all of the data needed for that modeling case and proceed with running SMOKE.

Once the SMOKE run has been completed, the contractor will load SMOKE QA reports created by SMOKE during processing. Additionally, the contractor will load as “external” datasets the model-ready input files. All of the additional QA steps that the contractor currently performs (without the EMF) while using SMOKE will be listed in the EMF, associated with these model-ready data. As the contractor completes each of these steps, the contractor will document the results in the EMF. This will allow EPA staff to ascertain at any time the status of the QA on a given modeling scenario and also have documentation to indicate that all of the QA steps have been done.

Typically, the first set of model-ready emissions data is not the last, especially for a brand-new modeling platform. Therefore, we anticipate that this process will be repeated one or more times. As the raw data are updated and improved, the EMF will track these changes. The QA steps will be performed again, though we expect that sometimes some steps will be able to be skipped on the second and later passes of the QA. The EMF will allow documentation of these skipped steps. Additionally, as missed problems are discovered, the EMF allows new QA steps to be added for datasets so those steps can be incorporated into future required or optional QA on similar data.

CONCLUSION

During the remainder of this fiscal year (through September of 2006), we will continue to develop the EMF in other ways. The Case Manager is the second major component to the EMF that will manage SMOKE runs. The version that we will complete this year will provide a mechanism to organize all SMOKE settings and input data for a SMOKE case, which covers using SMOKE for all sectors to create model-ready emissions data. Assuming resources can be found, we will implement a version of the Case Manager next year that will additionally provide a graphic user interface to SMOKE and perform the SMOKE runs. We hope to obtain the resources to make the EMF available to SMOKE users outside of EPA in 2007. Until EPA provides the EMF to the modeling community, it may be obtained directly from the University of North Carolina at Chapel Hill, Carolina Environmental Program.

As part of the overall EMF effort, we have also created three general-purpose tools for creating SMOKE inputs: the Speciation Tool, the Spatial Surrogate Tool, and EmisView. The Speciation Tool

can be used to create SMOKE speciation profiles from the SPECIATE4 database. We anticipate that this tool will be released later this fiscal year. This tool supports creating speciation profiles that support the ozone/PM chemical mechanism for the Community Multiscale Air Quality (CMAQ) model for ozone and PM (CB-IV, CB05, and SAPRC99) and for ozone, PM and toxics (CB-IVtx, CB05tx, and SAPRC99tx). Additionally, the Speciation Tool input files will be able to be managed by the EMF and the tool will eventually be run from the EMF. Another companion paper at this conference³ entitled “New Tools to Generate Spatial Surrogate and Speciation Profile Inputs to SMOKE,” provides more detail about this tool. The Spatial Surrogate Tool, which is also described in the same paper as the Speciation Tool, supports creating spatial surrogates for SMOKE without the need to understand or use a Geographic Information System (GIS). In the coming year, we anticipate that the integration of this tool with the EMF will be completed. EmisView is also described in the QA companion paper², and we will be working during this fiscal year to complete the initial integration of it with the EMF.

Additionally, OAQPS is also developing a replacement to the AirControlNET software [<http://www.epa.gov/ttn/ecas/AirControlNET.htm>] currently named the Control Strategy Tool (CoST). This tool will include a database of on-the-books control assumptions and control technologies that can be used to create control strategies for non-EGU point and stationary nonpoint sources. This tool is being built into the EMF so that it can leverage the Data Manager’s features for storing, editing, and tracking data changes. While initial plans are that this tool will be used internally at EPA, the developers plan to make it part of publicly released software in the future.

About a year ago, we reported our plans to create the EMF system. We are pleased to have most of the Data Management capabilities and supporting tools that we originally envisioned available to the OAQPS team. Although we are just beginning our use of the Data Manager, it is readily apparent to the team that the capabilities we planned a year ago are now becoming a reality that will greatly assist our efforts for the 2002-based modeling platform. We believe that our use of the Data Manager will result in improved data quality, staff and contractor performance, documentation of data, and transparency.

DISCLAIMER

The research presented here was performed in part under a Memorandum of Understanding between the U.S. Environmental Protection Agency (EPA) and the U.S. Department of Commerce’s National Oceanic and Atmospheric Administration (NOAA) and under agreement DW13921548. This work constitutes a contribution to the NOAA Air Quality Program. Although it has been reviewed by EPA and NOAA and approved for publication, it does not necessarily reflect their policies or views.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the contributions to the EMF Data Manager from the staff at the University of North Carolina at Chapel Hill, Carolina Environmental Program, including Conrad D'Cruz, Raghu Havaladar, Parthee Partheepan, Qun He, and Catherine Seppanen. Additionally, we would like to acknowledge the staff at UNC and Environ who contributed to the Surrogate Tool and Speciation Tool: Limei Ran at UNC and Greg Yarwood, Michele Jimenez, and Sunja Rao. Lastly, we would like to acknowledge others at EPA who have contributed their insights into the design of the EMF – Bill Benje, George Pouliot, Norm Possiel, and Dan Loughlin.

REFERENCES

- 1 Houyoux, M.; Strum, M., Possiel, N.; Benjey, W.G.; Mason, R.; Pouliot, G., Loughlin, D.; Eyth, A.; Seppanen, C. "EPA's New Emissions Modeling Framework", 14th Annual International Emission Inventory Conference, Las Vegas, NV, April 11-14, 2005.
- 2 Eyth, A.M; Partheepan, R. "Quality Assurance of Emissions Inventory and Modeling Data with the Emissions Modeling Framework and EmisView," 15th Annual International Emission Inventory Conference, New Orleans, LA, May 15 - 18, 2006.
- 3 Eyth, A.M.; Ran, L.; Yarwood, G.; Jimenez, M.; Rao, S. "New Tools to Generate Spatial Surrogate and Speciation Profile Inputs to SMOKE," 15th Annual International Emission Inventory Conference, New Orleans, LA, May 15 - 18, 2006.

KEYWORDS

Emissions Modeling Framework

Data Management

SMOKE

Quality assurance