The Emissions Inventory Database Application Component of the Air Quality Management Decision Support System for Beijing

Jo Ellen Brandmeyer, Eric Solano, Robert A. Zerbonia
RTI International, 3040 Cornwallis Rd., PO Box 12194, Research Triangle Park, NC 27709-2194
brandmeyer@rti.org

George Gao
PA Consulting Group, Washington, DC

Li Xin, Chen Tian, and Shi Aijun
Beijing Municipal Environmental Protection Bureau, Beijing, China

ABSTRACT

An integrated air quality management decision support system (AQMDSS) is being constructed to help the Beijing Municipal Environmental Protection Bureau (BMEPB) improve air quality for the 2008 Olympics and beyond. This region of China experiences poor air quality from multiple pollutants of concern, thus requiring a one-atmosphere approach to modeling. The AQMDSS will integrate tools for emissions inventory development and processing, industrial-scale air dispersion modeling, regional-scale photochemical modeling, and geospatial data analysis.

The emissions inventory database is the core of the AQMDSS. The database layout was constructed from a needs analysis of the models and processors that will use it. The National Emissions Inventory (NEI) Input Format version 3 (NIF3) was the starting point for the design. Some fields (e.g., country code) and tables (e.g., temporal factors, cross-reference tables) were added to support source models, emissions processors, geographic information system (GIS) analyses, and other models. Referential integrity and range checks were built into the database definition where appropriate.

A graphical user interface (GUI) allows users to enter and update data. Input forms reduce data entry errors by providing drop-down boxes and lookup capabilities for all codes. Also, construction of the forms supports the parent-child data relationships between tables in the database. The GUI supports a multilingual user interface. The initial version will be available in both English and Chinese. The application is also extensible, allowing more quality assurance (QA) tools to be easily added in the future.

INTRODUCTION

An AQMDSS is being created to help officials in Beijing, China, understand the underlying causes of Beijing's poor air quality and to simulate the effects of potential growth and control scenarios. Brandmeyer et al. (2004)[1] describes the overall design of this system. The BMEPB required a single, integrated system where the data are stored in a central database that state-of-the-science models and tools share. There are many advantages to using such an integrated system, including data consistency among models. Also, by storing data in a relational database management system, the data can be more easily analyzed using Standard Query Language (SQL) statements. The AQMDSS can be used for creating and analyzing emission inventories and executing air quality models in China and other countries, including the United States, for years to come.

The initial focus of the project was to design the centralized database, obtain and integrate the various models, and create a user interface in Chinese. The preferred models and tools were to be selected from existing models available from the U.S. Environmental Protection Agency (U.S. EPA) and in the public domain. Components that were not available from the U.S. EPA were selected from alternatives either in the public domain or licensed with source code under the GNU General Public License (GPL)[2] or a similar license.

The AQMDSS includes the following components:

- A centralized database for emissions inventory data
- Tools for entering and maintaining the emissions inventory for the entire Beijing modeling domain
- The Sparse Matrix Operator Kernel Emissions processing system (SMOKE) for processing the emissions inventory
- The Models-3/Community Multiscale Air Quality (CMAQ) system for regional air quality modeling
- The American Meteorological Society (AMS)/U.S. EPA Regulatory Model (AERMOD) for industrial-scale air dispersion modeling
- The Chemical Mass Balance (CMB) model for investigating source categories for particulate matter
- Tools for transferring data between the models and the database
- Data visualization tools
- GIS tools
- The Multimedia Integrated Modeling System (MIMS) modeling framework
- Additional data analysis tools.

An appropriate tool for entering and maintaining the emissions inventory could not be located. Therefore, we designed and built the requisite tool for this purpose.

The remainder of this paper discusses the design and operation of the emissions inventory portion of the AQMDSS. The Data Elements section discusses highlights of the database design. The Graphical User Interface section describes both the user interface and how its design minimizes data entry errors. The Expandability section outlines the steps required to add a new model and data tables to the AQMDSS. The final section describes the Operational Requirements of the emissions inventory portion of the AQMDSS.

DATA ELEMENTS

This section discusses the design of the relational database underlying the AQMDSS. The design supports the NIF3[3] data format, which was enhanced to support multiple countries, the models within the AQMDSS, and geospatial data analysis tools. Using relational principles, the database structure can be easily expanded to support the data requirements of other models and analysis tools.

The U.S. EPA provides the NIF3 specifications in Microsoft® Excel, Microsoft® Access, and XML formats. Although the NIF3 is inherently relational, the specifications do not explicitly include the multifield relations between the data tables. Also, basic codes (e.g., pollutant code) required for the emissions inventory data are provided separately from the format specifications. When a data provider (e.g., state agency) uploads emissions inventory data from its data storage system to the U.S. EPA, QA is performed on the data submission. The U.S. EPA notifies the data provider of any data errors. The data provider then edits and uploads the data again. This cycle is repeated until the data submission passes the QA step.

The AQMDSS uses a more proactive approach to data collection and maintenance. The database itself includes primary keys, unique keys, and foreign keys in all data tables. These keys reduce data errors by providing QA as the data are entered into the database. For example, when entering pollutant data, the pollutant code is a required field and the value of the pollutant code must exist in the table of pollutant codes.

Parent-child relationships between data tables are also enforced. For example, as shown in Figure 1, data for a point-source facility are stored in multiple data tables, each of which represents a level of detail. (Figure 1 shows the core data tables for point sources; supporting tables [e.g., temporal, speciation, cross-reference, codes] are omitted for clarity.) Data about the facility as a whole are stored in the Facility table at the top (parent) level. There can be one or more release points for a facility, one or more processes for a release point, etc. The structure of the GUI, as described in the next section, helps users to retain these data relationships and avoid orphaned records.

Maintaining an emissions inventory is an ongoing effort in which new data are gathered and existing data are improved. The AQMDSS database supports the process of data improvement through the use of a data reliability indicator. Because emission factors are a major source of uncertainty in the emissions inventory, the AQMDSS database includes a data reliability indicator for each emission factor. As shown in Table 1, these factors represent a hierarchy of confidence, where the data source that is most specific to the modeling domain is assigned a reliability indicator value of 1 (i.e., the highest indicator of reliability) and data sources that are increasingly less specific to the modeling domain are assigned increasing reliability values. When a new emission factor of greater confidence is obtained, the old record can be retained in the database for future analysis. A new record is entered with the improved emission factor and the appropriate reliability indicator. When data are extracted for use in the models and more than one data record exists, the one with the greatest confidence is extracted.

Measurement units provide another challenge for emissions inventory specialists. Data may be stored in metric or English units and at different scales (e.g., pounds or tons, grams or mg). All data, however, must be provided to the models in the units that each model expects. Therefore, units tracking is another feature of the AQMDSS. Data fields have units fields associated with them. When data are extracted from the database for use in a model, units conversions are automatically performed to put the data into the required units. A special table of units in the database stores the various units and conversion factors.

The database design includes some security features. Each user is assigned a user type within the user management portion of the AQMDSS. All users can look up data in the database, but data entry permissions vary by user type. Currently, users are designated by task, such as area source specialist, point source specialist, or visualization expert. These users are configured within the database, allowing each AQMDSS installation to define and maintain their own users and user types. When a record is added to the database, it is stamped with the current time stamp and the user ID of the person who entered the data. Also, when a record is updated, it is stamped with the current time stamp and user ID of the person who updated the record. The user ID and time stamp for both the creation and the last update of a record are maintained within the database.

Users can enter metadata (i.e., data about data) two ways. The first type of metadata is information specific to one datum or data record. When entering or updating data, the user can press a button that displays a special form for entering free-form notes that are attached to the data currently displayed. For example, the user may want to record why a datum was changed, its source, or its former value. The second type of metadata is information on an entire data file. Data file metadata is more structured, with separate fields to hold specific data. Metadata fields that support geospatial data sets include representativeness, geographic descriptors, and the time period covered by the data. More general metadata fields include a name for the data set, the source of the data, who obtained them, when they were obtained, where they are stored, their format, restrictions on their use, a formal reference for the data, and a statement of data quality. A reliability indicator code is also assigned to that data set.

GRAPHICAL USER INTERFACE

The AQMDSS includes a GUI for users to enter and update emissions inventory data. This section discusses several key features of this GUI, including enforcement of referential integrity within the database, support for multiple types of users, and a multilingual interface.

Referential Integrity

As discussed in the section on data elements, parent-child relationships exist within the database. One QA challenge is to ensure that orphaned records are not used in the emissions processor. For example, each point-source pollutant record must be linked to an active point source (see Figure 1). A point-source pollutant record without a valid point-source record is an orphaned record. A record can be orphaned when the code that should link it to its parent record is invalid.

The design of the GUI prevents orphaned records by supporting the referential integrity of the underlying database. This is accomplished by restricting how the user enters data. For example, a user cannot directly add a process record for a point source to the database. Instead, Figure 2 shows how the user must first select a facility and then the appropriate release point for that facility. A process record can then be added and is automatically linked to the release point and the facility records.

The GUI components were designed to be data driven. All GUI components are strictly graphical components, and data used to lay out or build the components are read directly from the database. No data are stored in the GUI structure, and changes in the databases will automatically be reflected in the GUI.

Multiple User Types

The construction of the GUI can be customized for each type of user from within the AQMDSS database. First, the types of users are defined by an administrator. Then, the administrator identifies the specific tables that each type of user can modify. Note that users add and edit data in the data tables, but cannot change the structure of the tables. Finally, each user is assigned a user type. When a user logs into the AQMDSS, the appropriate menu, based on the user type, is created and displayed.

Multilingual Interface

The multilingual user interface is another feature of the GUI. Two requirements of this project were that the user interface be provided in Chinese and that the developers provide support to the (Chinese) AQMDSS users. However, the developers are not literate in Chinese. To meet these requirements, the GUI has been programmed with a multilingual user interface. The program uses the locale set by the user to determine which language to use for display.

By using the internationalization/localization (i18n/l10n) features of the Java programming language, the user interface can be easily translated to other languages (e.g., Spanish). This programming technique removes all literal character strings (e.g., a frame title or the text on a button), places the strings in a text file, and replaces each string with an appropriate function call to the text file. The names of the text files include the language that the file contains. To support a new language, a new version of the text file must be created for that language, and the file name must include the appropriate designator for that language. No changes need to be made to the program’s source code.

We have taken this programming technique one step further. The GUI menus are dynamically built from information in the database, where the user type is assigned to each AQMDSS user. The labels for the various tables and fields are stored within the database as links to internationalized text files. The GUI, therefore, uses the locale of the user's environment in connection with the tables that the user can access to present the GUI in the appropriate language.

EXPANDABILITY

The AQMDSS was designed so that it can be easily expanded to include additional models, data elements, and user types. This type of expansion is supported through the relational database design, metadata-driven data maintenance, and extractors for models. This section discusses each of these three design features.

Relational Database Design

The relational design allows tables to be linked. The linkages are created through the use of primary and foreign keys (i.e., indices). A multifield key uses more than one column of data to define an identifier for a single row in a table. Figure 1 shows an entity-relationship diagram in which the fields composing the primary key are labeled PK and those composing a foreign key are labeled FK1. For example, the Facility table has a multifield primary key defined as Country + State + County + Facility ID. This primary key is a foreign key to the Release Point table. Because a facility can have more than one release point, the Release Point table's primary key is defined as Country + State + County + Facility ID + Release Point ID. Therefore, the tables Facility and Release Point have a one-to-many relationship.

Each table represents a certain level of specificity. The designation "Other Fields" in Figure 1 represents data values that are not part of the primary key. These fields provide additional attributes for a single record in that table. To add new data elements to the existing database, the user must examine the existing data structure to determine if there is already a data table that has the requisite primary key. If so, the user would decide whether it would be reasonable to add the new data elements to the existing table. If the existing table already contains numerous attributes or a large number of records that do not need to be populated with values for the new fields, then a new table can be created with a one-to-one relationship with the existing table.

One example where existing tables may need to be expanded is when an upgraded version of an existing model is released. This new model may add new parameters to one type of data. Because these new parameters need to be populated for all the existing data sources in the database, the new parameters should be added to the appropriate, existing data tables.

New tables may need to be created when a new model is added to the AQMDSS. If the new model will not be used with the full set of existing data, then appropriate related tables should be added. Data elements that are common between the existing data tables (e.g., facility name) and the new data should not be added. Instead, foreign keys should be used to create relationships between the new and existing data tables.

Metadata-Driven Data Maintenance

Whether new fields are added to existing data tables or new tables are added to the database structure, the new data elements need to be added to the data maintenance GUI. This task can be handled entirely within the database by adding the appropriate data to the metadata tables. The GUI code need not be modified. By adding the appropriate keys to the internationalized text files, the user interface for the new data elements will also support the same languages as the remainder of the AQMDSS.

After new data elements have been added to the database and the maintenance GUI, the models must be given access to those elements. If a model is being written to work directly with the database, the connector will be able to access the data. For existing or upgraded models, the new data will need to be provided to the model. The AQMDSS uses extractors for this purpose.

Model Extractors

When designing the AQMDSS system, one challenge was to handle models that are periodically upgraded. A key design decision was to not alter the models. Although it may be slightly more efficient to have the models read directly from the database, upgrading the model in the AQMDSS would then be more difficult. Instead, the AQMDSS includes numerous, well-defined processes called extractors.

The purpose of each extractor is to extract data from the database and put the values into the format expected by a specific model or processor. For example, the AQMDSS includes extractors that create the data files required by SMOKE. When all the data in a table are extracted, such as the list of SIC codes, then the only parameter required by the extractor is the name of the file to create. Other extractors need to know which data to extract from each table. These extractors use parameter values, such as time period, to extract the values needed for a scenario.

When a model is upgraded, if the data file formats are not changed, then the extractors do not need to be edited. If the formats of some data files are changed, then only the extractors that create those data files need to be edited. Similarly, when a new model is added to the AQMDSS, the extractors for the existing models do not need to be changed, even if data elements are added to existing data tables in the database. However, new extractors need to be created to extract the data required by the new model.

OPERATIONAL REQUIREMENTS

The AQMDSS has been constructed to minimize costs to users. It uses software from the U.S. EPA when possible. New components are created using generally available software tools that have liberal use licenses. Table 2 lists these tools, where each can be obtained, and its license.

The AQMDSS and its database were designed as a multiuser application. This means that the entire system can be used simultaneously by more than one person, limited only by the host hardware and operating system. The AQMDSS is being tested in both LINUX® and Microsoft Windows® computing environments.

CONCLUSIONS/SUMMARY

An integrated air quality management support system (AQMDSS) is being constructed to help officials in Beijing, China, improve Beijing’s air quality in time for the 2008 Olympics. The AQMDSS includes a centralized database with emissions inventory data, tools for managing data, and relevant air quality models from the U.S. EPA. The database design, which was based on the U.S. EPA’s NIF3 data format, includes enhancements to accommodate multiple countries, models, and data requirements. Important features in the AQMDSS include enforcement of primary keys for all tables, data reliability indicators for data quality control, tracking of measurement units, security management, and metadata support. The GUI is data driven and relies completely on the database. Its design includes support for referential integrity, and it can be customized for each type of user. The GUI supports multiple languages by implementing internationalization and localization rules. The AQMDSS can be easily expanded to include additional models and data elements and has been built to minimize costs to users by using publicly available software as much as possible.

REFERENCES

[1] Brandmeyer, J.E. et al. “Development of an Air Quality Management Decision Support System for Beijing China,” Presented at the 2004 Models-3 Conference, Chapel Hill, NC, October 18–20, 2004; available at http://www.cmascenter.org/html/2004_workshop/abstracts_presentations.html.

[2] See http://www.gnu.org/copyleft/gpl.html.

[3] See http://www.epa.gov/ttn/chief/nif/index.html.

KEY WORDS

Air quality models
Emissions inventory
Graphical user interface
Referential integrity
Decision support

ACKNOWLEDGMENTS

LINUX is a registered trademark of Linus Torvalds.
Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States and other countries.

TABLES

Table 1. Data reliability indicators for Beijing Municipal Environmental Protection Bureau's AQMDSS.

Reliability Indicator	Description
1	Beijing-specific
2	China-specific
3	Asia-specific
4	Europe-specific
5	Other
6	EPA default value

Table 2. Software components from sources other than the U.S. EPA.

Software Tool	How to Obtain	License
PostgreSQL object-relational database management system	www.postgresql.org	BSD license
pgAdmin III data dictionary tool for PostgreSQL	www.pgadmin.org	Artistic license
JDBC driver for PostgreSQL	jdbc.postgresql.org	BSD license
Java 2 Platform Standard Edition 5.0	java.sun.com	Sun license

FIGURES

Figure 1. Relationship of primary data tables for point sources.

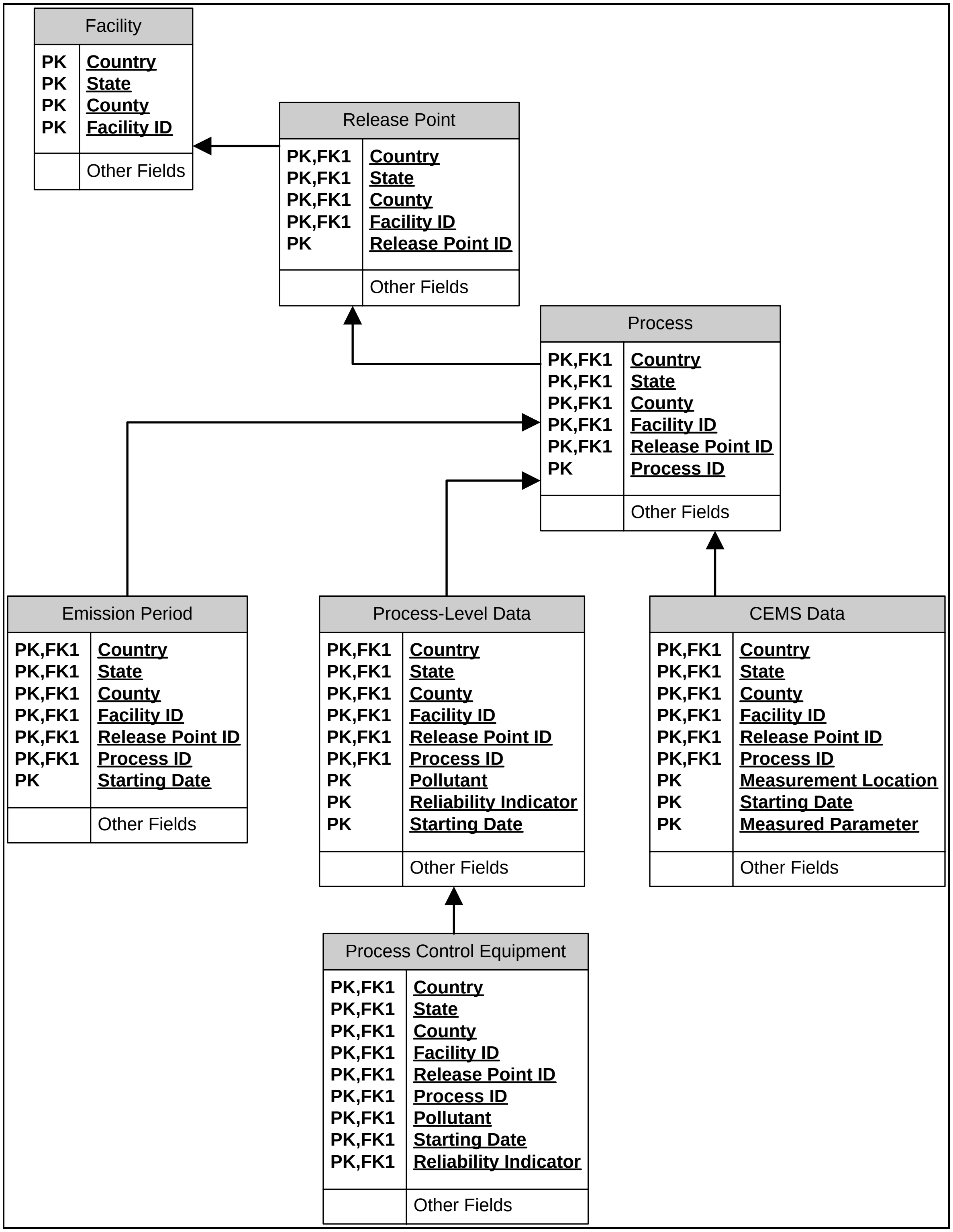

Figure 2. GUI screen showing organization of AQMDSS data tables for maintenance.