10. DEVELOPMENT OF TRIM COMPUTER FRAMEWORK

This chapter describes the computer framework that will be used for each of the TRIM modules and that is currently being implemented for TRIM.FaTE. Therefore, much of this discussion is specific to TRIM.FaTE, but it can be generalized to all of the TRIM modules. Additional information about these aspects of TRIM.FaTE Version 1.0 can be found in Appendix F of Volume I of the TRIM.FaTE TSD and in Fine et al. (1998a, 1998b).

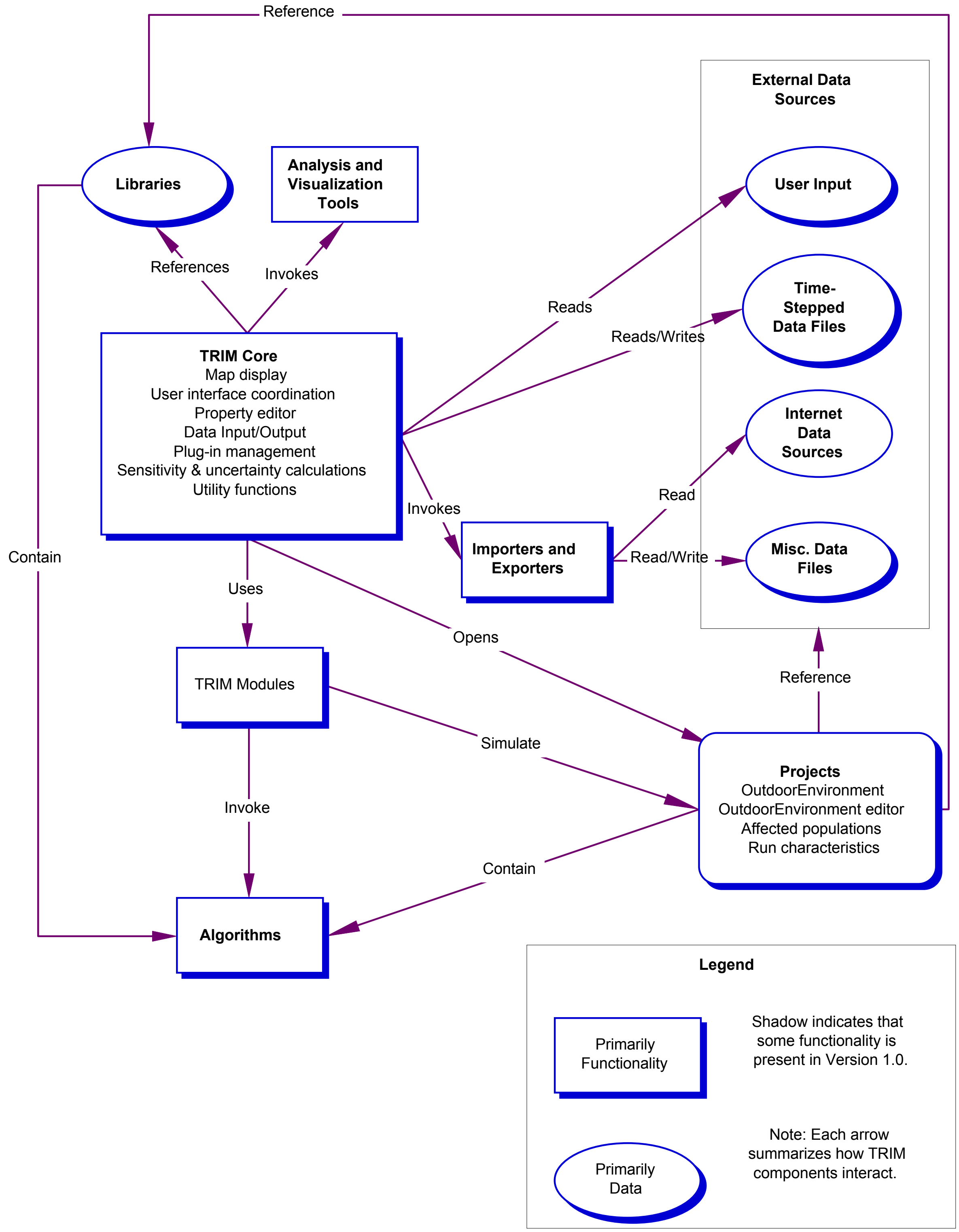
The development of TRIM.FaTE Version 1.0 began in 1998 and was completed in September 1999. Version 1.0 differs from the prototype in several ways. Specifically, Version 1.0 (1) is compatible with operating systems beyond Microsoft Windows, such as UNIX, (2) provides improved management of multiple modeling scenarios, and (3) is easier to use and more reliable. Similar to Prototype V of TRIM.FaTE, TRIM.FaTE Version 1.0 provides users with the following options:

- Define the parameters of each specific assessment, including time period, geographic region, pollutants, environmental media, and populations of interest;
- Choose appropriate pollutant fate and transport algorithms for use in assessments;
- Select modeling parameters, including emissions sources, characteristics of the environment (*e.g.*, air temperature, soil permeability), and simulation time step (*e.g.*, hourly, daily);
- Identify and access input data sets, and identify and create output data sets;
- Execute the assessment; and
- Export results.

10.1 ARCHITECTURE

As shown in Figure 10-1, the TRIM computer system architecture is complex but flexible, allowing it to be applied in developing each of the different TRIM modules. The architecture components used to describe TRIM are classified as those that primarily provide (1) functionality (rectangles), and (2) data (ovals). However, each of the components except for external data sources provide both functionality and data. The architectural components that have been implemented to some degree in Version 1.0 are depicted with shadows. This figure is designed to represent the relationships within the TRIM computer framework, rather than the data flow within the system. Therefore, the word along an arrow forms a sentence where the verb on the arrow connects the two architecture components at the end of an arrow. For example, in the upper left hand corner of the figure, the TRIM Core “invokes” Analysis and Visualization Tools. Each of the TRIM components shown in Figure 10-1 are described below.

Figure 10-1
TRIM Computer System Architecture



10.1.1 TRIM CORE

The TRIM Core component primarily provides services required by multiple architectural components or integrates those components. The following items are included in the Core.

- A mapping tool shows volume elements and associated information, such as predicted chemical concentrations, and is based on an off-the-shelf software component that provides some GIS-like capabilities. The mapping tool allows users to view geospatial data from external sources, such as soil type layers generated by a GIS and stored in a SHAPE file, with an overlay of TRIM information.

- A simple graphical user interface allows the user to invoke TRIM modules, such as TRIM.FaTE, and that maintains lists of open windows.

- A property editor enables users to edit and view property values, where a property value describes an attribute (*e.g.*, molecular weight) of an entity that is simulated by the module, such as a chemical or compartment or volume element. Examples of attributes for which property values are used include air temperature, scavenging coefficients, and chemical reaction rates.

- A management system allows user to plug in data importers and exporters.

- An analysis feature calculates sensitivity, uncertainty, and variability of outputs using TRIM modules (Note: this may not be supported in Version 1.0).

- Utility functions, such as routines that assist with data storage and retrieval, are used by TRIM modules.

10.1.2 PROJECTS

Projects in TRIM are used to store all information pertinent to an individual assessment. A project contains “scenarios,” where each scenario contains a description of the outdoor environment being simulated, populations being studied, and model parameters, such as the simulation time step. Each project also displays the information it contains and allows the user to change that information. In some cases, the information display and manipulation functions of a project rely on a TRIM Core functionality, such as the property editor.

10.1.3 TRIM MODULES

Each TRIM module, such as TRIM.FaTE, is a component that allows for simulation or analysis. Where required, modules also provide specialized graphical user interfaces that support their functionality. Version 1.0 includes only the TRIM.FaTE module. Future TRIM versions will include additional TRIM modules.

10.1.4 LIBRARIES

A substantial amount of relatively static information is required to conduct assessments of multimedia chemical fate and transport and subsequent exposures and effects on selected populations. For instance, static information includes the measured properties of chemicals that change infrequently or the boundaries of a study region that might stay constant for years. Because of the static nature of this information and because a large amount of static information may be needed for a single assessment, users can store such information in TRIM libraries. Users can then easily reuse selected information from a library in future projects. Changes may be made to the library over time to ensure that the most current science is used in assessments. However, when a user creates a project that accesses information from a library, a copy of the information is made to protect the project from future changes to the library.

The TRIM.FaTE module uses a number of chemical fate and transport algorithms that compute chemical transfer coefficients between and chemical transformation coefficients within compartments. As new chemicals, ecosystems, and relationships are studied, new algorithms will be required. In anticipation of this need, TRIM.FaTE has been designed to allow users to add algorithms. The algorithms are stored in libraries and can be applied to various projects, as designated by the user. Specifically, a user can manually assign algorithms stored in libraries to links or can request that TRIM.FaTE assign applicable algorithms based on the compartments that are connected by a link. For instance, some algorithms might only be applicable for transfer from surface water to fish. Even when TRIM.FaTE assigns algorithms, the user can review the assignments and make changes before the simulation starts. Before or after a simulation, the user can export the simulation scenario and its results (if available) to a set of HTML files. These HTML files show which algorithms were used for each link and the formulation of each algorithm.

10.1.5 EXTERNAL DATA SOURCES, IMPORTERS, AND EXPORTERS

Given the diversity of potential applications of TRIM, data required to address those applications, and formats used for storing that data, it is difficult to construct a computer framework that provides all potentially required capabilities. The TRIM architecture addresses this issue in several ways.

The architecture allows the user to add data importers and exporters in a relatively easy manner, as needed. Data importers read non-TRIM data sets and create and/or set appropriate TRIM objects and properties. For instance, Version 1.0 contains a data importer that can read a text file describing volume elements and can create the corresponding elements in a TRIM project. Another data importer can read a textual description of algorithms, compartments, chemicals, and sources and can create the corresponding objects in a TRIM library. Data exporters can write TRIM configurations and results in a format that is suitable for use by another computer program or for interactive review. Version 1.0 can export the configuration of a simulation scenario and its results to HTML files and simulation results to a text file that can be imported by Microsoft® Excel. Future data importers and exporters could provide many other capabilities. Examples include reading data produced by a GIS (*e.g.*, SHAPE files) and interpolating values to TRIM volume elements, writing results in a format that could be further processed by a GIS, importing information directly from a web site or database, and transferring results to a statistical package that is executing concurrently with TRIM. To provide additional flexibility, future versions of TRIM may allow knowledgeable users to apply data importers and exporters that users develop without modifying TRIM.

The TRIM.FaTE module, in specific, allows users to provide environmental data in binary files that can be read as needed by a TRIM.FaTE simulation. This streamlines the use of large data sets, such as hourly temperatures or concentrations over a 30-year period. Binary files can also be used for storing TRIM.FaTE results. The TRIM Core supports reading data from and writing data to file formats that are based on the Environmental Decision Support System/Models-3 Input/Output Applications Programming Interface (I/O API) (Coats 1998). The I/O API format can be easily read and written from several programming languages, is platform-independent, is suitable for large data sets, is self-describing (*i.e.*, contains information about variables and time periods contained in the file), and is computationally efficient.

10.1.6 ANALYSIS AND VISUALIZATION TOOLS

Version 1.0 does not include any analysis or visualization tools. Instead, simulation results can be easily exported to Microsoft® Excel or other analysis packages. In the future, TRIM will include some analysis and visualization capabilities and may allow users to develop and plug in additional capabilities.

10.2 IMPLEMENTATION APPROACHES AND TECHNOLOGIES

The TRIM is being developed using an object-oriented approach. There has been much discussion in the software engineering literature, such as Booch (1993), on the benefits of this approach, including increased software extensibility, reusability, and maintainability. The essence of object-oriented software development is that concepts, such as a volume element, are represented as a unit that contains internal data (*e.g.*, the boundaries of a volume element) and operations on the data (*e.g.*, computation of volume), and that one class of objects (*e.g.*, volume element with vertical sides) can be a specialization of another class of objects (*e.g.*, volume element). Being able to specialize classes of objects allows general functionality to be shared by several specialized classes. The TRIM's representation of the outdoor environment (with volume elements that contain compartments) and the development of associated graphical user interfaces are well suited for an object-oriented treatment.

The TRIM is being developed in an iterative manner. The major components and responsibilities of a class of objects are understood before implementation, but some details may need to be resolved as implementation proceeds. Prior to implementation, graphical user interface mock-ups and significant new capabilities are shown to potential users. During implementation, the design is modified as needed. This user-oriented development approach helps highlight potential problems before undesirable approaches become embedded in the system. Furthermore, the object-oriented, open-ended structure of TRIM is intended to make future changes and additions a relatively simple process.

For Version 1.0 of TRIM, simpler and/or more reliable approaches were used in preference to faster and/or less resource-intensive approaches. In cases where simple approaches did not have adequate performance or significantly limited the potential for future changes, more complex approaches were used. Operations that caused noticeable speed or resource problems were optimized as time and resources permitted.

The TRIM computer framework and TRIM.FaTE module have been developed primarily, but not entirely, in the Java programming language. Some parts of TRIM.FaTE, such as the differential equation solver, and other TRIM modules, such as TRIM.Expo, ultimately will be implemented in the FORTRAN programming language. Advantages of using Java include the following.

- Java code is portable across different hardware and operating systems. This is especially important for graphical user interfaces, which will comprise a large fraction of the TRIM code and which can be difficult to develop for multiple platforms.

- Java offers a combination of speed of development, dependable system behavior, and support for object-oriented designs.

- Java is supported by multiple vendors, often leading to competitive pressures to improve development tools. In addition, it reduces the likelihood that one vendor's product strategy or financial problems will negatively affect TRIM development.

- Java provides built-in support for multithreading (*i.e.*, allowing multiple operations to proceed simultaneously) and networking (*i.e.*, communicating with software on remote computers, such as extracting simulation properties from a web-based data repository).

The disadvantages of using Java as the primary programming language for TRIM include the following.

- Programs written in Java typically execute more slowly than programs written in C++ or BASIC. However, as the technologies for compiling and executing Java programs advance, the execution time for Java programs should decrease.

- Fewer plug-in components (*e.g.*, mapping tools) and libraries (*e.g.*, matrix manipulation) are available for Java than are available for languages such as C++ or BASIC on Windows. However, the number of plug-in components available for Java is continuing to grow.

- Java development tools are not as mature (*e.g.*, fewer tools, lower performance, greater probability of system errors) as tools for other languages, but that situation is improving.

10.3 USING TRIM.FaTE VERSION 1.0

Version 1.0 of TRIM.FaTE as completed in September 1999. This section provides a general discussion of how a user would set up and run a simulation using Version 1.0.

After starting TRIM.FaTE Version 1.0, the user can create a new project or library or open an existing project or library. A library populated with objects must be created by the user before any meaningful work can be performed with a project. Note that when TRIM.FaTE is distributed in the future, some pre-loaded libraries will be included. From the library window (shown in Figure 10-2), the user can choose to create new or examine existing algorithms, chemicals, compartments, point sources, or property types. TRIM.FaTE Version 1.0 also allows the user to import objects from text files rather than creating objects from the Graphical User Interface (GUI).

Properties are an important concept in TRIM.FaTE Version 1.0 because they store information about the objects in the system. Examples of properties for a chemical include melting point, vapor pressure, and molecular weight. Examples of properties for an algorithm include the receiving compartment type, the sending compartment type, and whether it transforms a chemical. Each property references a property type that defines the type of data (*e.g.*, real number, date and time, true or false), a default value, a description, units, and for numeric data types recommended minimum and maximum values. In TRIM.FaTE Version 1.0, a GUI component, the Property Editor (shown on the right sides of Figures 10-2 and 10-3), is used throughout the system to add properties to objects and to view and edit the values of properties. For some properties in Figure 10-1, the value field contains "<See Below>." These are special properties for which the value is a formula. An example of the use of formulas as properties is specifying how transfer factors are calculated for algorithm objects.

After creating a library that contains the algorithms, chemicals, compartments, and point sources to be used in the simulations, a project can be created with scenarios that will run the simulations. New projects are created with one scenario by default, and additional scenarios can be added as needed. Generally, the scenarios in a project are related in some manner. Libraries are attached to projects and serve as sources of objects for the scenarios. Typically, after creating a scenario, the user sets its properties. These include the begin and end times for and the time step for the simulation. After setting the properties, volume elements can be imported from a text file into the scenario. In a later version of TRIM.FaTE, a GUI will be available for defining and viewing volume elements.

The outdoor environment window (shown in Figure 10-3) is organized as a set of tabbed panes that allow the user to define the sources, chemicals, compartments, links, and algorithms that comprise the outdoor environment. The general procedure for populating the outdoor environment is to copy objects from libraries into the scenario's outdoor environment, and then to customize the objects as needed. Abiotic compartments are automatically added to the outdoor environment when the volume elements are imported, whereas biotic compartments can be manually added and deleted from the Compartments tab. A function called "Smart Add" is available to intelligently add biotic compartments to the abiotic compartments based on their properties. The volume elements, compartments, and links in the outdoor environment are displayed in an outline form that can be expanded and collapsed to display varying levels of detail. Links can be created manually using the Links tab or automatically using the "Smart Link" function. With "Smart Link," links are created between adjacent or co-located compartments if algorithms that connect their compartment types exist in the project's libraries. Algorithms on links can be viewed and added or removed manually from the Algorithms tab.

After the properties for the scenario are set and the sources, chemicals, compartments, links, and algorithms are assigned to the outdoor environment, it is possible to run a simulation. The Verify button on the scenario window can be used before running the simulation to ensure that all necessary information is available (*i.e.*, all properties needed by the simulation have values). The Run button on the window is used to start the simulation. After the simulation is executed, the results can be exported to HTML and to text files that can be imported by Microsoft® Excel or another spreadsheet program.

Figure 10-2
Library Window of TRIM.FaTE Version 1.0

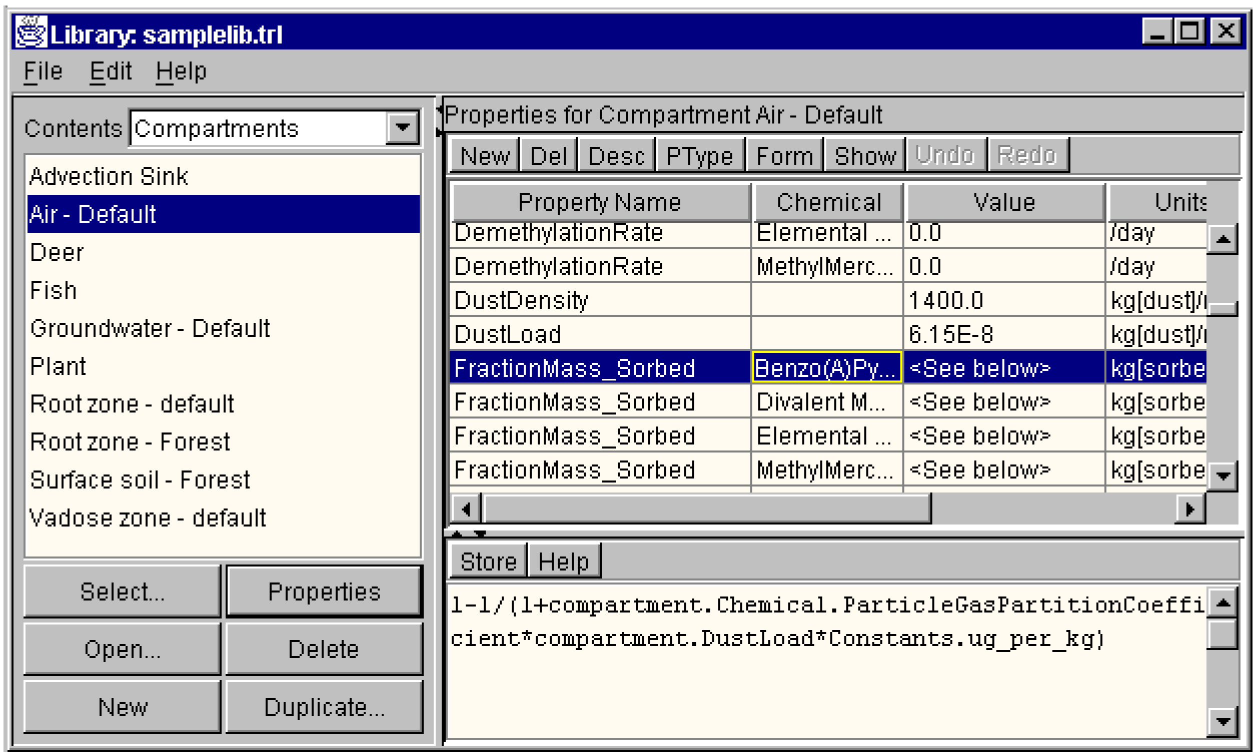
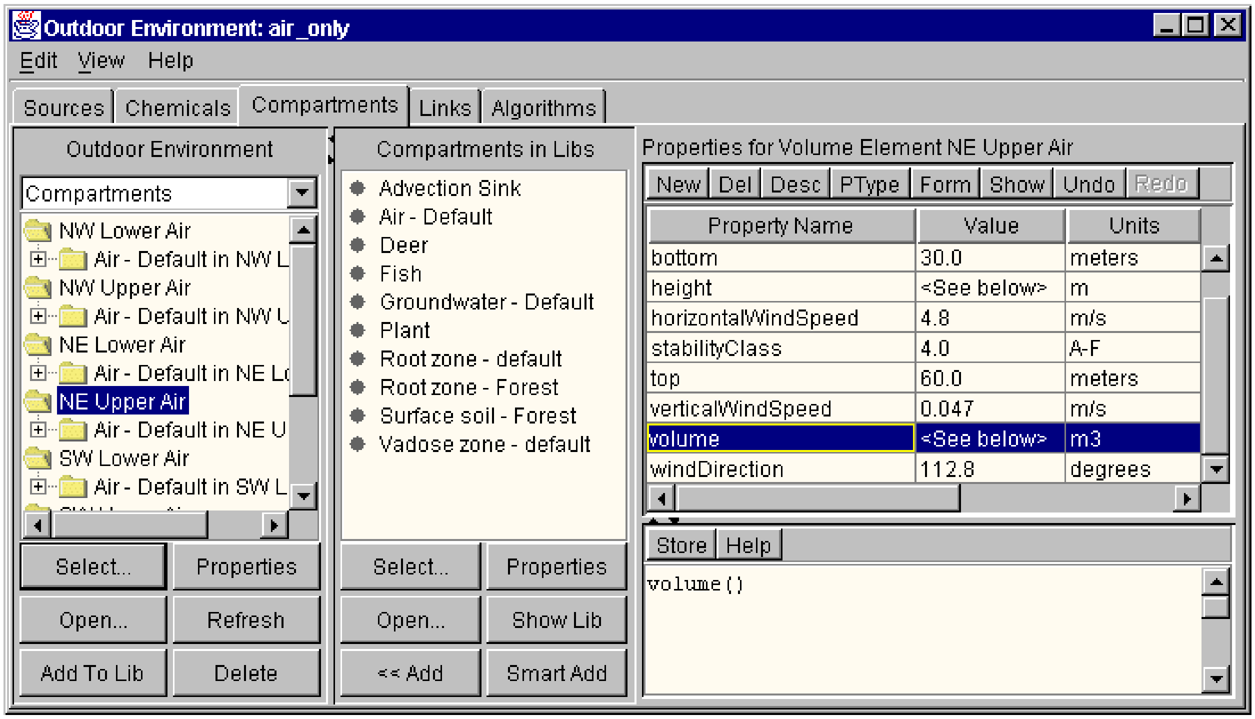


Figure 10-3
Outdoor Environment Window of TRIM.FaTE Version 1.0



10.4 IMPLEMENTATION STATUS

Version 1.0 provides all of the functionality listed at the beginning of this chapter. The software is currently being evaluated to identify problems and to gain confidence in the system. Major additions that will be implemented in the future include data analysis tools, sensitivity and uncertainty studies, and the display of geographic data.

[This page intentionally left blank.]