US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

DOCUMENTATION FOR THE FRAMES-HWIR TECHNOLOGY SOFTWARE SYSTEM, VOLUME 6: MULTIMEDIA MULTIPATHWAY SIMULATION PROCESSOR

Project Officer
and Technical Direction:

Mr. Gerard F. Laniak
U.S. Environmental Protection Agency
Office of Research and Development
National Environmental Research Laboratory
Athens, Georgia 30605

Prepared by:

Pacific Northwest National Laboratory
Battelle Boulevard, P.O. Box 999
Richland, Washington 99352
Under EPA Reference Number DW89937333-01-0

U.S. Environmental Protection Agency
Office of Research and Development
Athens, Georgia 30605

October 1999

US EPA ARCHIVE DOCUMENT

DISCLAIMER

This report was prepared as an account of work sponsored by the U.S. Environmental Protection Agency. Neither Battelle Memorial Institute, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC06-76RLO 1830

This document was printed on recycled paper.
(9/97)

US EPA ARCHIVE DOCUMENT

ACKNOWLEDGMENTS

A number of individuals have been involved with this effort. Mr. Gerard F. Laniak of the U.S. Environmental Protection Agency (EPA), Office of Research and Development, National Environmental Research Laboratory, Athens, Georgia, provided the overall technical direction and review throughout this work. This report was prepared by the Pacific Northwest National Laboratory[1] (PNNL) staff of Karl Castleton, Regina Lundgren, John Buck, Gariann Gelston, Bonnie Hoopes, John McDonald, Mitch Pelton, and Randall Taira. Additional PNNL staff supporting this effort include Wayne Cosby, Nancy Foote, Kristin Manke, Jill Pospical, Debbie Schulz, and Barbara Wilson. Useful inputs were provided by many U.S. EPA individuals working on the Hazardous Waste Identification Rule, including Messrs. Barnes Johnson, Stephen Kroner, and David Cozzie, and Drs. David Brown, Robert Ambrose, Zubair Saleem, Donna Schwede, and Sharon LeDuc, among many others.

[1]Operated by Battelle for the U.S. Department of Energy under Contract DE-AC06-76RLO 1830.

US EPA ARCHIVE DOCUMENT

Summary

The U.S. Environmental Protection Agency (EPA) is developing a comprehensive environmental exposure and risk analysis software system for agency-wide application. The software system will be applied to the technical assessment of exposures and risks relevant to the Hazardous Waste Identification Rule (HWIR). The software system adapted to automate this assessment is the Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES), developed by the Pacific Northwest National Laboratory (PNNL). The process used to develop the FRAMES-HWIR Technology Software System includes steps for requirements analysis, design, specification, and development with testing and quality assurance composing a critical portion of each step. This report documents that process for one of the key components of the system: the Multimedia Multipathway Simulation Processor (MMSP). This processor will

- perform a simulation that represents a multimedia, multipathway source, transport, exposure, and risk assessment
- execute the following computational modules:
  - source modules (aerated tank, land application unit, landfill, surface impoundment, and waste pile)
  - fate and transport modules (air, vadose zone, aquifer, watershed, and surface water)
  - foodchain modules (terrestrial, farm, and aquatic)
  - human exposure and risk modules
  - ecological exposure and risk module.
- execute the proper sequence and manage data flow through the various computational modules
- receive input through the Site Simulation Files (SSF), which contain data to control the simulation and most of the data required for a complete assessment
- pass information necessary for the different computational modules to execute (that is, provide the name and location of the data files)
- incorporate FRAMES-HWIR Technology Software System specifications that link the computational modules into a simulation for the HWIR assessment
- provide input in an appropriate format to the Exit Level Processor (ELP) through the Global Results Files (GRF)
- direct the storage of varying degrees of data, depending on user need [defined by the user through the System Management Screen of the System User Interface (SUI)]
- terminate processing and return control to the SUI when errors cause computational modules to abnormally terminate
- report any MMSP-specific errors to the SUI
- provide information on time elapsed for the processing of each computational module
- allow testing as a stand-alone processor, independent of the other FRAMES-HWIR Technology Software System processors

- contain computational modules that have been unit-tested to ensure they meet requirements.

The five key design elements of the MMSP are

1. SSF - files in flat-ASCII (American Standard Code for Information Interchange) format, consisting of a directory of Data Group files that group data by function (for example, site layout, source, pathway). By using this directory, a computational module can quickly read only the Data Groups of interest, ignoring the rest. The complete set of input variables required for an execution of the MMSP is uniquely named and organized into these Data Groups.

2. Computational modules - representations of the components of the risk assessment process that consist of a model with pre- and/or post-processors, as needed. All computational modules follow the FRAMES-HWIR Technology Software System specifications, either directly for newly developed computational modules or by the addition of a pre- and/or post-processor, and/or recoding the existing computational module. Following these specifications assumes the computational modules access all databases and data files required without guidance from the MMSP, other than passing file names and locations. Computational modules meet resource allocations of 64 Mb of RAM, 250 Mb of disk space, and an execution time of about 1 second on a stand-alone PC (assuming a Pentium [586] 200 MHz PC-based machine).

3. MMSP Module Execution Manager - a component that provides each computational module with the file name and location of the input data it requires in the SSF and potentially GRF Data Groups. The Module Execution Manager also implements the source, transport, and exposure/risk modules in the proper firing sequence and manages storage activities for the MMSP.

4. HWIR input/output dynamic link library (HWIRIO.DLL) - a group of programs that encompasses several subroutines, including those for initialization, input, output, and finalization. In general, a computational module uses the HWIRIO.DLL to read input variables from the SSF, read modeled input variables from the GRF, and write outputs to the GRF.

5. GRF- files in flat-ASCII format that consist of a directory of Data Group files with results for each computational module, by source type, waste concentration, chemical, and iteration. The GRF represent a complete set of output variables generated by an execution of the MMSP, uniquely named and organized into Data Groups. The GRF are used as input to the ELP.

All components of the MMSP have been tested with respect to their specific requirements. The MMSP Module Execution Manager was tested as a unit to ensure it met requirements and performed as expected. Test cases evaluated the capability of the Module Execution Manager to link various computational modules in sequence, time module execution, and report errors and warnings correctly. The MMSP Module Execution Manager passed its tests. In addition, development of the MMSP Module Execution Manager followed a quality assurance program designed to ensure the processor met EPA expectations. Further information on the MMSP computational modules will be published by the EPA as separate reports that constitute appendices of this report.

US EPA ARCHIVE DOCUMENT

Acronyms and Abbreviations

ASCII	American Standard Code for Information Interchange
ELP	Exit Level Processor
EPA	U.S. Environmental Protection Agency
FRAMES	Framework for Risk Analysis in Multimedia Environmental Systems
GRF	Global Results Files
HWIR	Hazardous Waste Identification Rule
HWIRIO.DLL	HWIR input/output dynamic link library
Mb	megabyte
Met	meteorological
MHz	megahertz
MMSP	Multimedia Multipathway Simulation Processor
PC	personal computer
PNNL	Pacific Northwest National Laboratory
RAM	random access memory
SSF	Site Simulation Files
SUI	System User Interface

US EPA ARCHIVE DOCUMENT

Contents

US EPA ARCHIVE DOCUMENT

Figures

Tables

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

1.0 Introduction

The U.S. Environmental Protection Agency (EPA) is developing a comprehensive environmental exposure and risk analysis software system for agency-wide application. The software system will be applied to the technical assessment of exposures and risks relevant to the Hazardous Waste Identification Rule (HWIR). The HWIR is designed to determine quantitative criteria for allowing a specific class of industrial waste streams to no longer require disposal as a hazardous waste (that is, to *exit* Subtitle C) and to allow disposal in Industrial Subtitle D facilities. Hazardous waste constituents with values less than these exit criteria levels would be reclassified as nonhazardous wastes under the Resource Conservation and Recovery Act.

The software system adapted to automate this assessment is the Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES), developed by the Pacific Northwest National Laboratory (PNNL). The FRAMES-HWIR Technology Software System consists of a series of components within a system framework (Figure 1.1). The process used to develop the FRAMES-HWIR Technology Software System includes steps for requirements analysis, design, specification, and development with testing and quality assurance composing a critical portion of each step.

This report discusses one of the major components of the system: the Multimedia Multipathway Simulation Processor (MMSP). This processor represents the framework around the multimedia multipathway exposure and risk assessment capability required for the FRAMES-HWIR Technology Software System. The MMSP contains two primary subcomponents: 1) computational modules that represent the source of contamination, environmental fate and transport, foodchain, and exposure/risk to humans and ecological receptors; and 2) the MMSP Module Execution Manager, which manages the sequential execution of the required computational modules and other functions for each simulation.

The interrelationships between input/output data files and data transfer, computational modules, the Module Execution Manager, and the System User Interface (SUI) are schematically illustrated in Figure 1.2. As shown in the figure, the MMSP receives input from the Site Simulation Files (SSF), as well as user input through the SUI. Following directions in this input, the MMSP Module Execution Manager provides the names and locations of the files needed to execute the appropriate computational modules. The MMSP Module Execution Manager ensures the computational modules execute in the appropriate sequence to assess risk for a particular simulation. Computational modules write results to the Global Results Files (GRF). These results are used by the other computational modules and the Exit Level Processor (ELP) to calculate potential waste concentrations by site and source type.

This report includes information on the requirements of the MMSP Module Execution Manager and the computational modules and design elements necessary to meet those requirements. It also discusses testing plans, testing results, and the quality assurance program for the MMSP Module Execution Manager. Specifications for the MMSP Module Execution Manager and the computational modules are described in *Documentation of the FRAMES-HWIR Technology Software System, Volume 8: Specifications*. Further information on the MMSP computational modules will be published by the EPA as separate reports that constitute appendices to this report. References cited in the text are listed in Section 6.0. Appendix A provides additional details on the testing program for the MMSP Module Execution Manager. Other components developed by PNNL are described in companion documents as listed in the reference list; the system itself is documented in a summary report entitled *Overview of the FRAMES-HWIR Technology Software System*.

US EPA ARCHIVE DOCUMENT

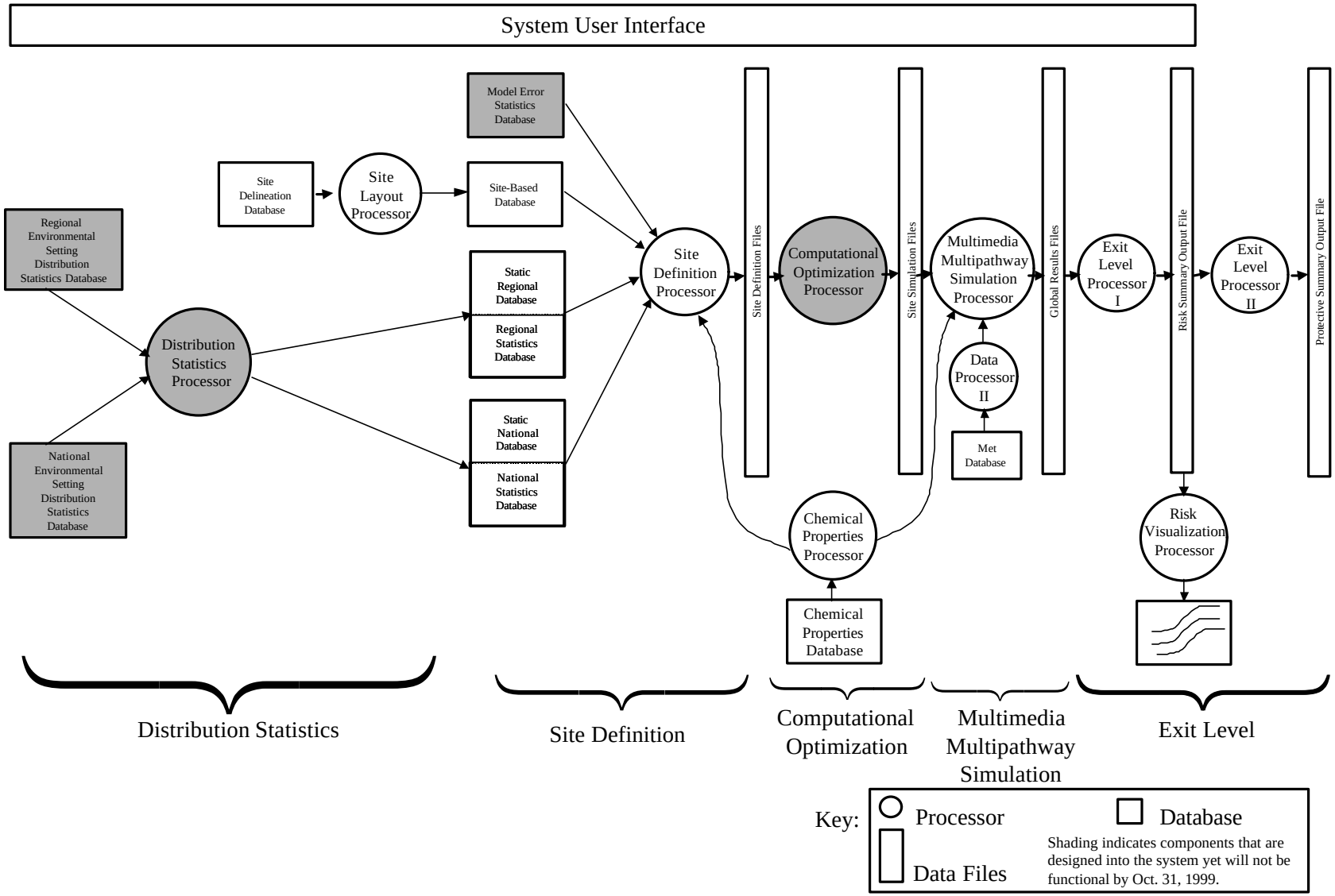


Figure 1.1 Overview of the FRAMES-HWIR Technology Software System

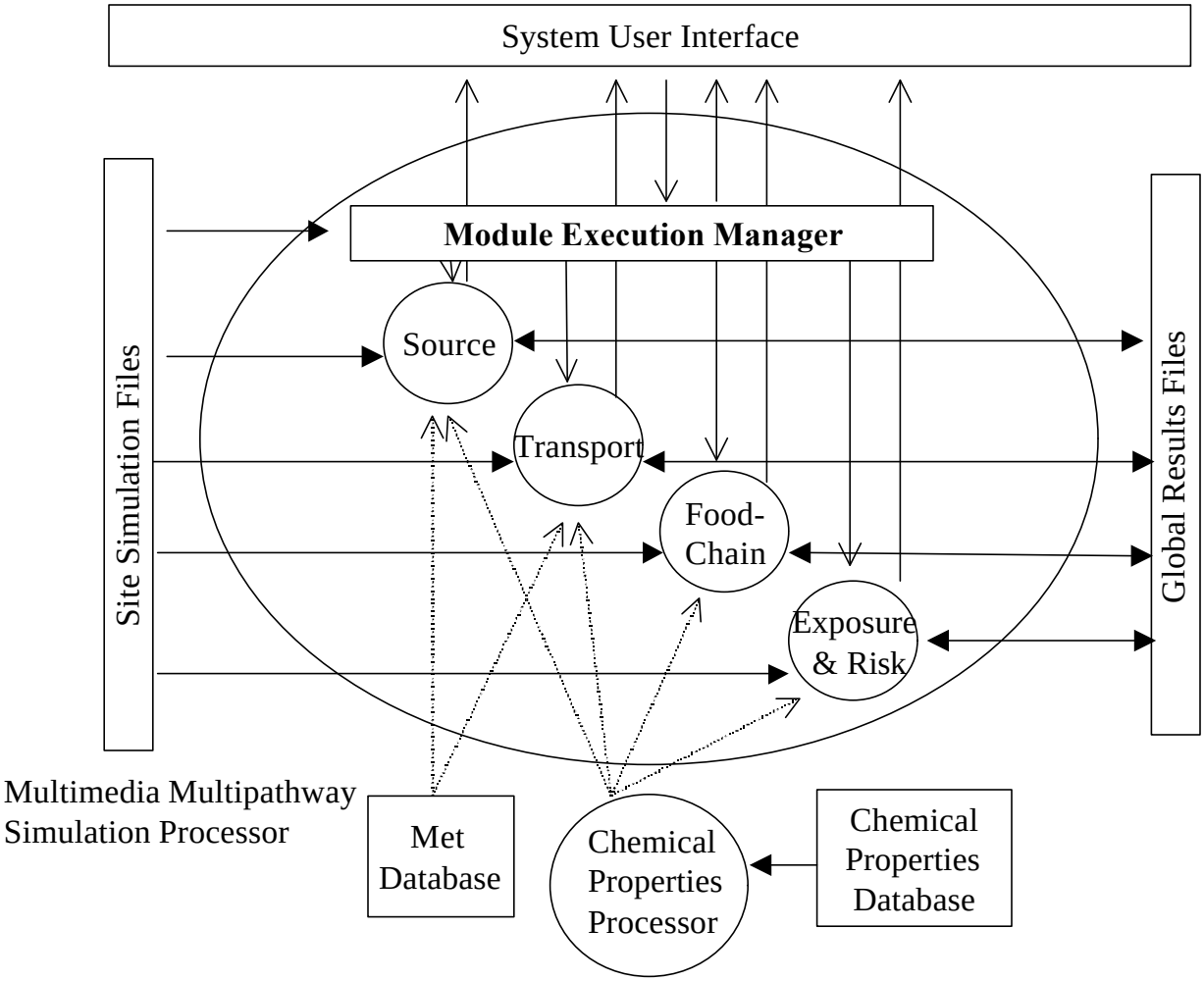


Figure 1.2 Details of the Multimedia Multipathway Simulation Processor
(Dashed lines indicate input from complementary components that are outside the MMSP)

US EPA ARCHIVE DOCUMENT

2.0 Requirements

Requirements are characteristics and behaviors that software must possess to function adequately for its intended purpose. The MMSP represents the framework around the multimedia multipathway exposure and risk assessment capability required for the FRAMES-HWIR Technology Software System and comprises a Module Execution Manager and computational modules. The MMSP Module Execution Manager manages the sequential execution of the required computational modules for each simulation. The computational modules reflect the individual components of the risk assessment process for the HWIR assessment. A computational module is defined as a model and the pre-postprocessors as necessary for the HWIR assessment. The following module types are included:

- Source modules (aerated tank, land application unit, landfill, surface impoundment, and waste pile)
- Fate and transport modules (air, vadose zone, aquifer, watershed, and surface water)
- Foodchain modules (terrestrial, farm, and aquatic)
- Human exposure and risk modules
- Ecological exposure/risk module.

In summary, this processor will

- Perform a simulation that is represented by a multimedia, multipathway source, transport, exposure, and risk assessment
- Execute the following computational modules:
  - source modules (aerated tank, land application unit, landfill, surface impoundment, and waste pile)
  - fate and transport modules (air, vadose zone, aquifer, watershed, and surface water)
  - foodchain modules (terrestrial, farm, and aquatic)
  - human exposure and risk modules
  - ecological exposure/risk module.
- Execute the proper sequence and manage data flow through the various computational modules
- Read input from the header Site Simulation Files (SSF), which contain data to control the simulation and most of the data required for a complete assessment
- Provide information necessary for the different computational modules to execute (that is, provide the name and location of the data file)
- Incorporate FRAMES-HWIR Technology Software System specifications that link the computational modules into a simulation for the HWIR assessment
- Provide input in an appropriate format to the ELP through the GRFs

US EPA ARCHIVE DOCUMENT

- Direct the storage of varying quantities of data, depending on user need (defined by the user through the System Management Screen of the System User Interface [SUI])
- Terminate processing and return control to the SUI when errors cause computational modules to abnormally terminate
- Report any MMSP-specific errors to the SUI
- Provide information on time elapsed for the processing of each computational module
- Be able to be tested as a stand-alone processor, independent of the other FRAMES-HWIR Technology Software System processors
- Contain computational modules that have been unit tested to ensure they meet their requirements.

The following subsections describe in further detail the input, output, and scientific requirements of the MMSP.

2.1 Input Requirements

The MMSP primarily receives input via the SSF. The MMSP Module Execution Manager manages the flow of that information through the various computational modules. The computational modules then read their specific input data file(s) in the SSF from information provided to them by the MMSP Module Execution Manager. (This information consists of the name and location of the data file containing the data.)

Although the SSF provide the primary input to the MMSP, several of the computational modules within the MMSP require additional files that would be computationally inefficient to transfer through the SSF. For example, it would be inefficient to transfer to the SSF all meteorological data for a station located near a site and then transfer these data through the system to the computational modules that utilize the data. Instead, the MMSP is designed to process the meteorological station reference data through the different computational modules and to require the appropriate computational modules to read the detailed data from the referenced database.

Another set of input files that do not necessarily travel through the SSF are those of the Chemical Properties Database. Static chemical data are processed by the Site Definition Processor and transferred through the system to the SSF. Chemical property information that is module-dependent is passed directly to these modules, as they are responsible for reading that data from the database with the help of the Chemical Properties Processor.

US EPA ARCHIVE DOCUMENT

2.2 Scientific Requirements

A primary purpose for the MMSP Module Execution Manager is that it must manage the sequential execution of each computational module required for a specific site processed within the FRAMES-HWIR Technology Software System. Management of the execution of the computational modules implies providing and permitting access to information needed by individual computational modules (for example, file names/locations). Such management also implies the MMSP can recognize software problems that cause computational modules to terminate abnormally, as well as the warnings associated with suspect calculations, and execute explicit protocols (that is, terminate entire run and report symptoms to user).

Assumptions for the computational modules within the MMSP include the following:

- All computational modules follow the specifications outlined in *Documentation of the FRAMES-HWIR Technology Software System, Volume 8: Specifications* (see Section 6.0). Newly developed models can follow the specifications directly; models developed before the FRAMES-HWIR Technology Software System effort can meet specifications by the addition of a pre- or post-processor, and/or recoding of the existing computational module. Following these specifications assumes the computational modules can access all required databases and data files without guidance from the MMSP, other than for passing file names and locations.

- All computational modules have been unit tested by the module developer by undergoing a suite of test cases to confirm the computational module meets its requirements.

- All computational modules meet the expectations described in *Documentation for the FRAMES-HWIR Technology Software System, Volume 9: Software Development and Testing Strategies* (see Section 6.0, Volume 6) for quality assurance and software development.

- Computational modules cannot consume all of the memory, disk space, and time resources available to the FRAMES-HWIR Technology Software System. The resource allocations for a computational module are 64 Mb of RAM, 250 Mb of disk space, and an execution time of about 1 second on a stand-alone PC (assuming a Pentium [586] 200 MHz PC-based machine). The 1-second execution time for each module is assumed, although not necessarily guaranteed.

2.3 Output Requirements

The MMSP must provide output to the ELP through the GRF. The MMSP is designed to allow saving any or all files used in the exposure and risk assessment per site. The computational modules within the MMSP are required to create the GRF for each simulation processed. The GRF include information related to the estimation of risk per site, per source type, per chemical, and per level of input waste concentration. Thus, for a simulation, the MMSP retains in the GRF key risk outputs from the computational modules.

The MMSP passes processor-specific error files, as well as warning and error files associated with each computational module, to be processed by the SUI. These warning and error files reflect the status of the MMSP Module Execution Manager and the computational modules executed during the simulation. These files are written in flat-ASCII file format, as described in *Documentation of the FRAMES-HWIR Technology Software System, Volume 8: Specifications* (Section 6.0).

US EPA ARCHIVE DOCUMENT

The MMSP also provides output information regarding the time taken by each computational module to process. This information is provided through the GRF. Computational module execution timing is needed to ensure that the FRAMES-HWIR Technology Software System conducts the HWIR assessment in a manner that meets user needs for information availability.

2.4 Future Directions

In the future, the MMSP could create a database that would contain all data required to conduct *what if* analyses to support the HWIR assessment. These analyses would be conducted based on questions and comments received by reviewers of the HWIR assessment exit level risk results. At a minimum, the primary output (for example, contaminant fluxes, contaminant concentrations, dose, and impacts) would be retained in this data file. Additional data developed by specific computational modules could also be included for the analysis of specific results.

US EPA ARCHIVE DOCUMENT

3.0 Design Elements

Design elements are strategies for meeting requirements. The MMSP is designed to meet the requirements identified in Section 2.0. Key to meeting those requirements is the interface between the various computational modules associated with the MMSP. The following subsections describe input to the MMSP, computational module design, implementation of the MMSP, and output from the MMSP. Included in the module design section (3.2.1) is information on the HWIR Input/Output Dynamic Link Library (HWIRIO.DLL), a set of programs used by the FRAMES-HWIR Technology Software System to facilitate communication between components.

3.1 Multimedia Multipathway Simulation Processor Input

The MMSP receives input from the SSF, as well as input from the user through the SUI. The SSF are in flat-ASCII format and consist of a directory of Data Group files. This directory allows a computational module to quickly read only those Data Groups it needs, ignoring the rest. The complete set of input variables required for an execution of the MMSP is uniquely named and organized into these Data Groups. Within a Data Group, each variable is uniquely identified by name and indexes. Variable values may be scalar or multidimensioned. Scalar variables are accessed by name only. Multidimensioned variables are accessed by name and indexes (from 1 to 6). Variable data types include integer, real, and character strings.

The SSF contain the following Data Groups:

- header
- site layout
- chemical properties
- source (aerated tank, surface impoundment, landfill, waste pile, land application unit)
- air
- vadose zone
- aquifer
- watershed
- surface water
- aquatic foodchain
- terrestrial foodchain
- farm foodchain
- ecological exposure
- ecological risk
- human exposure
- human risk.

The specific variables within each of these groups are listed in *Documentation for the FRAMES-HWIR Technology Software System, Volume 8: Specifications* (see Section 6.0).

The two Data Groups of most importance to the MMSP Module Execution Manager are the Header Data Group and the Site Layout Data Group. The Header Data Group provides information on the

US EPA ARCHIVE DOCUMENT

name of the site being simulated, the time and date of the creation of the Site Definition Files (from which the SSF were created), the names of the various computational modules and processors, and the dates and times computational modules and processors were executed. The Site Layout Data Group provides the information on which pathways (computational modules) to execute. For example, the Site Layout Data Group might indicate no pathway to the vadose zone and groundwater for an aerated tank, so the MMSP Module Execution Manager would not instruct those computational modules to execute. The connections between modules were designed to match the HWIR Assessment Strategy (Marin and Saleem 1997).

The MMSP also receives information from the SUI, including chemicals to be considered, iterations within the simulation, and storage levels and locations. For additional information on the SUI, consult *Documentation for the FRAMES-HWIR Technology Software System, Volume 2: System User Interface* (see Section 6.0). Additional information on input specific to individual computational modules is described in the subsections to Section 3.2.

3.2 Module Design

A computational module is a component within the MMSP that comprises some combination of model, pre-processor, and post-processor. The computational modules reflecting the individual components of the risk assessment include source modules, transport modules, foodchain modules, and exposure and risk modules. Figure 3.1 indicates the general interactions between these computational module types within the MMSP. The following subsections describe input, execution, and output design features of the MMSP modules.

3.2.1 Module Input

Much of the information needed to allow the computational modules to execute is passed from the MMSP Module Execution Manager through the HWIRIO.DLL. Instead of being compiled with the main program, this file is dynamically linked with the program that uses it during program execution. The set of such files is somewhat comparable to the library routines provided with programming languages such as FORTRAN, C, and C++. The design of the HWIRIO.DLL encompasses several subroutines, including those for initialization, input, output, and finalization. In general, a computational module uses the HWIRIO.DLL to read input variables from the SSF, read modeled input variables from the GRF, and write output to the GRF.

The MMSP design and implementation require that each computational module obtain input from specified SSF Data Groups and, possibly, specified GRF Data Groups. Computational modules use the HWIRIO.DLL subroutines to read all SSF and GRF Data Groups. The MMSP Module Execution Manager provides each computational module with the file name and location of the input data it requires in the SSF and potentially in the GRF Data Groups. Two exceptions to this provision are the meteorological data and, for some computational modules, the chemical properties data. If a computational module needs meteorological data (observations of wind speed, direction, and stability over a defined time period), that computational module is expected to read the Met database using methods designed and implemented by EPA. If the computational module requires chemical properties data that are not provided through an SSF or GRF, the computational module is expected to use the Chemical Properties Processor to read from the Chemical Properties Database.

US EPA ARCHIVE DOCUMENT

3.2.2 Module Execution

Computational modules representing each of the components of the risk assessment process were formulated outside of the FRAMES-HWIR Technology Software System development effort. Although the MMSP Module Execution Manager guides the sequence of execution and provides the computational modules with names and locations of input files, the computational modules operate as *black boxes* in the MMSP. Thus, these modules do not require the MMSP to provide additional information on how to execute them.

Some of the computational modules used for the MMSP incorporate legacy models, that is, models that were developed for previous applications. Often, legacy modules require some modification to work within the FRAMES-HWIR Technology Software System, at a minimum by the addition of pre- or post-processors. Legacy models that are currently planned for use in the MMSP include ISC-ST for the Air Module, EXAMS for the Surface Water Module, and EPA-CMTP for Vadose Zone and Aquifer Modules.

The module developers are expected to reformat and reorder data to meet the specific needs of their computational module. The reformatting and reordering of data within a computational module can be done by one of three methods:

1. Use pre-processors and/or post-processors without modifying the legacy model.
2. Modify the legacy model to directly read specified files using shared subroutines.
3. Use a combination of processors and model modifications.

These options are shown in Figure 3.2. Note if pre- or post-processors are created for a computational module, a *batch* file also is required as part of the computational module. This batch file serves as the single entry point for execution of that computational module.

3.2.3 Module Output

The MMSP is designed to allow for varying degrees of output storage, addressing varying degrees in quantity (that is, temporary versus permanent files). Each computational module is responsible for correctly interpreting the data storage level, defined by the user through the SUI. With a data storage level of zero, the MMSP Module Execution Manager copies only the GRF to permanent space, deleting any temporary files the computational module might have created during its execution. If a computational module is given a data storage level greater than zero, the computational module is responsible for determining what files in addition to the GRF are saved. If a computational module leaves a file in temporary space and the data storage level is greater than zero, the MMSP Module Execution Manager copies that file into permanent space.

Computational modules are also expected to write their outputs to a given location using the subroutines in the shared HWIRIO.DLL. The GRF output required by each computational module is detailed in *Documentation for the FRAMES-HWIR Technology Software System, Volume 8: Specifications* (see Section 6.0).

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

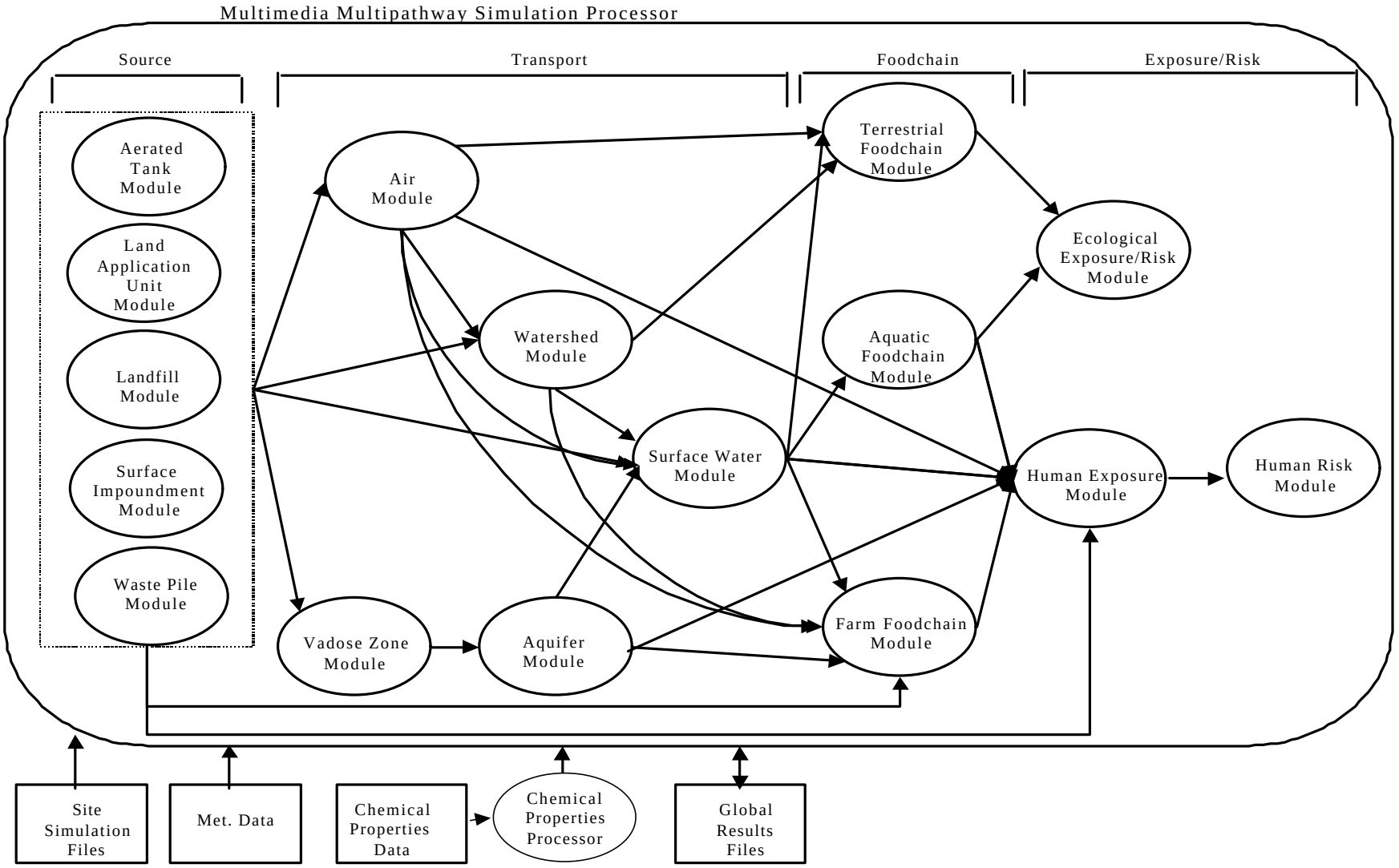


Figure 3.1 Interactions Within the Multimedia Multipathway Simulation Processor

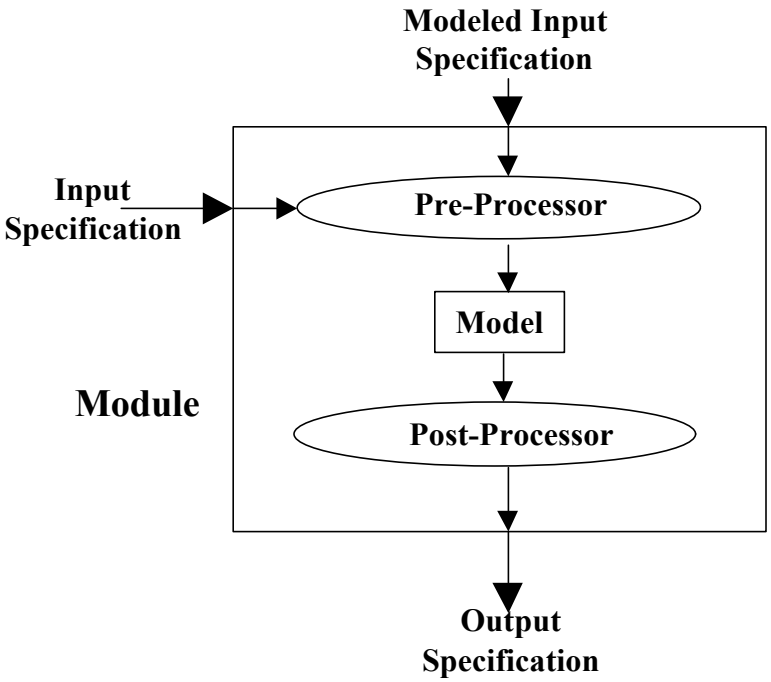


Figure 3.2 How Legacy Models Connect with the FRAMES-HWIR Technology Software System

Computational module errors and warnings are reported to the MMSP, which in turn reports them to the SUI via the HWIRIO.DLL. The error and warning files are written in free-format ASCII. Errors are defined as anything that causes the computational module to terminate abnormally. Through the HWIRIO.DLL, a computational module creates an error file at the start of execution and deletes it at the end of execution (provided no errors are produced). If an error is produced, the HWIRIO.DLL closes the error file and MMSP terminates execution of the computational module. The opening/creation and closing/deletion of this file is handled automatically with the OpenGroups and CloseGroups Subroutines provided in the HWIRIO.DLL.

Through the HWIRIO.DLL, the computational modules also create a warning file at the start of execution and delete it at the end of execution, provided no warnings are produced. If a warning is produced, the computational module writes the warning to the file and continues processing. The opening/creation and closing/deletion of this file is handled automatically with the OpenGroups and CloseGroups Subroutines provided in the HWIRIO.DLL. If the algorithm successfully continues to the end with neither errors nor warnings, the error file and the warning file are deleted.

3.3 Multimedia Multipathway Simulation Processor Implementation

The MMSP receives the SSF from either the Site Definition Processor or the Computational Optimization Processor and, in combination with the SUI information, executes the appropriate computational modules. This predetermined set of computational module executions form a simulation, which results in estimates of the impacts to human and ecological receptors (see Figure 3.1). Results from each computational module are exported to the GRF, where the ELP processes them.

US EPA ARCHIVE DOCUMENT

The MMSP responds to the following call arguments passed to it via input from the SUI through the HWIRIO.DLL:

SSF_Dir	; Directory of SSF Data Groups (string)
GRF_Dir	; Directory of GRF Data Groups (string)
Header_SSF	; Header SSF Data Group name

The Header_SSF Data Group will contain at least the following arguments that are used by the MMSP.

Realization	; Current outer loop count (integer)
Site	; Current site being simulated (integer)
SourceType	; Current source type (string)
ChemId	; Current chemical identification (string)
CwIndex	; Current concentration level (integer)
StorageLevel	; Storage level (integer)
Time	; Time when SSF was created (string)
Date	; Date when SSF was created (string)

Site and outer loop in these call arguments refer to looping programs associated with the statistical strategy of the simulation. The site loop is the iteration within the FRAMES-HWIR Technology Software System that loops through all sites or locations associated with the HWIR Assessment Strategy; it is also known as the site assessment loop. The outer loop is the iteration within the FRAMES-HWIR Technology Software System that allows for the random selection of measurement and sampling errors associated with the key risk assessment variable. The outer loop cycles through parameter distributions considered in the Monte Carlo simulation.

Source type in the call arguments refers to the aerated tank, land application unit, landfill, surface impoundment, and waste pile modules. A waste management unit may be made up of several source types, and a source type may contain more than one source. For example, a facility outside Chicago (the waste management unit) may contain two landfills (one source type, two sources) and a collection of aerated tanks (second source type, five sources).

Computational module call arguments are set through a Windows/DOS environment variable. The environment variable is called ARGUMENTS and is set by the MMSP Module Execution Manager at run time. The ARGUMENTS variable is changed by the MMSP for each separate computational module execution.

A general description of the MMSP algorithm is as follows:

1. For selected computational modules
2. If Module Input data have changed
3. Get System Time
4. Execute the Module
5. Get System Time and Compute Difference and Add Difference to Modules Total Time
6. End if
7. If data storage above 1 (from SUI)
8. Copy files in temporary space to permanent space

US EPA ARCHIVE DOCUMENT

9. Else If data storage equal 1 (from SUI)
10. Clean up un-needed files
11. End if
12. End loop on selected modules

3.4 Multimedia Multipathway Simulation Processor Output

Each MMSP module produces a specific GRF Data Group that can be used as input to the ELP. Computational modules are expected to write their respective GRF Data Groups using the FRAMES-HWIR Technology Software System data specifications. The groups are written to a location given by call arguments using the subroutines in the HWIRIO.DLL.

The writing of temporary files (any file created by the computational module that is not an .SSF or .GRF file) is acceptable, but the files should be deleted before exiting the computational module. This deletion process allows the remaining files to match the appropriate storage level given to the computational module in the call argument.

The GRF consist of results for each computational module by source type, waste concentration, chemical, and iteration. The GRF represent a complete set of output variables generated by an execution of the MMSP, uniquely named and organized into Data Groups. Within a Data Group, each variable is uniquely identified by name and indexes. Variable values may be scalar or multi-dimensional. Scalar variables are accessed by name only. Multi-dimensioned variables are accessed by name and indexes (from 1 to 6). Variable data types include integer, real, and character strings. The specifications for the GRF are in *Documentation for the FRAMES-HWIR Technology Software System, Volume 8: Specifications* (see Section 6.0). The GRF contains the following Data Groups:

- source
- site layout
- air
- vadose zone
- aquifer
- watershed
- surface water
- aquatic foodchain
- terrestrial foodchain
- farm foodchain
- ecological exposure
- ecological risk
- human exposure
- human risk.

US EPA ARCHIVE DOCUMENT

4.0 Testing Approach and Results

As noted, the MMSP encompasses the risk assessment process of the FRAMES-HWIR Technology Software System. Associated with this processor are several subcomponents, including the following items:

- Computational modules developed by other organizational teams supporting the technology development effort - these modules comprise models and associated pre- and/or post-processors.

.

- Module Execution Manager - this subcomponent manages the sequence of execution, storage of processed data, and other functions of the processor.

- Meteorological database - this database, developed and populated by the EPA, supplies input to the computational modules.

- Chemical Properties Database and Processor - this database, developed and populated by the EPA, supplies input to the computational modules through the Chemical Properties Processor.

- HWIRIO.DLL - this input/output dynamic link library is used by the computational modules, as well as the MMSP and other system processors, to provide the specifications for call arguments to access and provide data.

- SSF - these files provide input to the MMSP and the computational modules.

- GRF - these files are produced by the computational modules within the MMSP. Included in these files are warning and error information generated by the computational modules and error information generated by the MMSP Module Execution Manager that, in turn, are echoed to the SUI for appropriate response by the user.

This section describes the type of testing conducted for the MMSP Module Execution Manager, summarizes the requirements on which testing was based, and describes test cases and the results of their implementation. Additional information related to testing can be found in Appendix A, including specifications of the Site Layout Data Group, which is critical to testing the MMSP Module Execution Manager, and procedures for setting up test cases.

4.1 Type of Testing

Software testing can be performed at the unit and system levels. Unit testing evaluates individual components in isolation from other components (for example, the Module Execution Manager in isolation from the FRAMES-HWIR Technology Software System). System testing evaluates the performance of groups of components functioning together, data communication between the components comprising the system (also called integration testing), and the overall performance of the system (for example, testing the functioning of the MMSP within the FRAMES-HWIR Technology Software System). The tests described in Section 4.3 address unit testing (that is, of the MMSP Module Execution Manager rather than the computational modules contained within the MMSP).

US EPA ARCHIVE DOCUMENT

4.2 Summary of Requirements

Requirements for the MMSP Module Execution Manager are described in Section 2.0. of this document. These requirements were reworded for the list in Table 4.1 of fundamental requirements suitable for testing. Note these requirements comprise only those that relate to the MMSP Module Execution Manager. Requirements for the computational models are discussed in the module documentation, which will be published by EPA at a later date as separate reports that constitute appendixes to this report. Note that the format of the SSF and GRF, along with the defined Data Groups in these files and variables within those groups, is described in the document entitled *Documentation of the FRAMES-HWIR Technology Software System, Volume 8: Specifications*.

To ensure the MMSP Module Execution Manager meets the six requirements listed in Table 4.1, eight test cases were developed to check performance. Table 4.2 shows the relationship between these requirements and the test cases described in Section 4.3.

Table 4.1 Fundamental Requirements for Testing the Multimedia Multipathway Simulation Processor Module Execution Manager

Requirement Number	Requirement
1	Following cues in the Site Layout Data Group and Header SSF, execute in appropriate sequence the various computational modules (Aerated Tank, Land Application Unit, Landfill, Surface Impoundment, and Waste Pile Source Modules; Vadose Zone, Aquifer, Air, Watershed, and Surface Water Fate and Transport Modules; Aquatic, Farm, and Terrestrial Foodchain Modules; Human Exposure and Risk Modules; and an Ecological Exposure and Risk Modules). (For a description of the Site Layout Data Group and other components mentioned in this requirement, refer to Section 3.0 of this document.)
2	Provide the name and location of the data file to allow computational modules to access appropriate information from either the SSF or GRF.
3	Be capable of working with the OpenGroups subroutine of the HWIRIO.DLL (see *Volume 10: Software Development Kit* of the FRAMES-HWIR Technology Software System Documentation) .
4	Store varying degrees of data (as defined by the user through the SUI, Level 0–store minimum data, and Level 1–store all data) used in the exposure and risk assessment per simulation per site.
5	Report any processor errors to the SUI through the HWIRIO.DLL.
6	Time the execution of computational modules and provide statistics to the SUI to determine which modules consume the most processing time.

US EPA ARCHIVE DOCUMENT

4.3 Test Cases

All tests are conducted under Windows® 95, which represents the required operating system for implementing the HWIR Assessment Strategy. Because computational modules require the name and location of the SSF and GRF to execute properly, all test cases evaluate the ability of the MMSP Module Execution Manager to provide this information to them. In addition, because data storage level must be passed as a command for the Module Execution Manager to process, most test cases evaluate the ability of the Module Execution Manager to store minimal data (storage Level 0, which saves only the GRF output of each computational module into the GRF Directory). Test Case MMSP_07 evaluates the ability of the Module Execution Manager to store all data produced by the computational modules into a permanent directory.

Table 4.2 Relationship Between Test Cases and Fundamental Requirements

		Test Case Name (MMSP_xx)							
		01	02	03	04	05	06	07	08
Requirement	1	x	x	x	x	x	x	x	
	2	x	x	x	x	x	x	x	
	3	x	x	x	x	x	x	x	x
	4							x	
	5								x
	6	x	x	x	x	x	x		

The datasets used for the SSF, GRF, and the Site Layout Data Group described in the following subsections are extremely large and are not replicated in this document. However, file copies are available upon request. The procedure for creating test case input files is described in Appendix A.

Test cases MMSP_01 and MMSP_06 focus on testing the ability of the MMSP Module Execution Manager to execute the appropriate computational modules in the proper sequence. Data in the MMSP must flow from computational modules representing the sources of contamination, to those representing the environmental fate and transport of the contamination. In turn, these data provide input to computational modules representing foodchain uptake of the contamination and ultimately those representing exposure and risk to humans and ecological receptors (see Figure 1.2). Within each of these categories of computational modules (source, fate and transport, foodchain, and exposure/risk), data from one computational module often serve as input for another. The test cases were designed to ensure that each possible sequence of data flowing from one computational module to another was followed at least twice. Table 4.3 shows how Test Cases MMSP_01 through MMSP_06 test each potential module sequence within the MMSP.

US EPA ARCHIVE DOCUMENT

Because no computational modules were available during unit testing, a surrogate module (*dummymod.exe*) was used to represent all computational modules. The type of computational module *dummymod.exe* represents at a given instance in processing is shown by the type of data read and copied. Hence, when air input files are passed and copied, the air module is assumed to be operating, even though a review of the processing data would reveal that only *dummymod.exe* was operating.

In addition, all tests represent potential scenarios for which a user might want to determine risks using the FRAMES-HWIR Technology Software System. Test cases MMSP_01 to MMSP_06 also test the ability of the MMSP Module Execution Manager to time module execution. Minor differences (2 seconds or less) can be attributed to the slowing of the computer clock when large data files are passing.

4.3.1 MMSP_01

4.3.1.1 Description and Rationale

The MMSP Module Execution Manager must follow cues in the Site Layout Data Group, as well as input from the user through the SUI to ensure the computational modules are executed in the proper sequence. If the computational modules do not execute sequentially, the risk assessment does not complete accurately. The MMSP computational modules can be executed in a number of sequences, depending on the scenario described by the user (see Figure 3.1). These scenarios can range from simple to highly complex. This test case evaluates the ability of the MMSP Module Execution Manager to execute the simplest of these scenarios, that of a single source type (land application unit), resulting in direct human exposure and risk (Figure 4.1; note that this calculation is performed by using the Farm Foodchain Module human exposure parameters).

4.3.1.2 Input Data

Data needed for input to the MMSP include the suite of SSF (with the Site Layout Data Group) and GRF. A Site Layout Data Group has the file structure shown in *Documentation of the FRAMES-HWIR Technology Software System, Volume 8: Specifications.*

In addition, this test case requires information to simulate user input. The commands to run this test case can be found in the file *runMMSP-01.bat*, which passes the following commands to the MMSP Module Execution Manager:

```
del grf\*.grf
MMSP ssf grf hd111111.ssf
```

where the first two commands (del grf*.grf >) tell the MMSP Module Execution Manager to delete any previous GRF, ssf and grf are the locations of the necessary SSF and GRF directories, and *hd111111.ssf* is the name of the header file. This file contains information to direct the MMSP Module Execution Manager to the Site Layout Data Group, which also serves as input to this test case. The Site Layout Data Group for this test case is *slla111111.ssf*. This file directs the MMSP Module Execution Manager to execute the computational modules in the proper sequence described in Section 4.3.1.3.

US EPA ARCHIVE DOCUMENT

Table 4.3 Test Cases Related to Module Linkages

Module Sequence	**01**	**02**	**03**	**04**	**05**	**06**
Source to Air			x	x	x	x
Source to Farm Foodchain	x	x			x	x
Source to Surface Water					x	x
Source to Vadose Zone		x		x	x	x
Source to Watershed					x	x
Air to Farm Foodchain			x	x	x	x
Air to Human Exposure			x	x	x	x
Air to Surface Water			x	x	x	x
Air to Terrestrial Foodchain			x	x	x	x
Air to Watershed			x	x	x	x
Vadose Zone to Aquifer				x	x	x
Watershed to Farm Foodchain			x	x	x	x
Watershed to Terrestrial Foodchain			x	x	x	x
Watershed to Surface Water			x	x	x	x
Aquifer to Surface Water				x	x	x
Aquifer to Human Exposure				x	x	x
Aquifer to Farm Foodchain				x	x	x
Surface Water to Farm Foodchain			x	x	x	x
Surface Water to Aquatic Foodchain			x	x	x	x
Surface Water to Terrestrial Foodchain			x	x	x	x
Surface Water to Human Exposure			x	x	x	x
Aquatic Foodchain to Ecological Exposure			x	x	x	x
Aquatic Foodchain to Human Exposure			x	x	x	x
Farm Foodchain to Human Exposure	x	x	x	x	x	x
Terrestrial Foodchain to Ecological Exposure			x	x	x	x
Ecological Exposure to Ecological Risk			x	x	x	x
Human Exposure to Human Risk	x	x	x	x	x	x

US EPA ARCHIVE DOCUMENT

4.3.1.3 Expected Results

The MMSP Module Execution Manager runs all modules without errors or warnings. Computational modules in process are approximated by reading and copying their SSF input files. The SSF for the Land Application Unit Module is read first, followed by those for the Farm Foodchain Module, Human Exposure Module, and the Human Risk Module. No other input files are used. Results are written to the GRF location specified by the MMSP Module Execution Manager (in this case, the GRF directory). In addition, no errors or warnings remain after processing. Hand-timing results approximate MMSP Module Execution Manager timing results within 2 seconds.

4.3.1.4 Procedure

The tester uses a stop watch or other clock capable of determining time in seconds. Using the Windows® Explorer, the tester double-left-mouse-clicks on the file *runMMSP-01.bat*, noting the time. The MMSP Module Execution Manager accesses the simulated SSF, including the appropriate header file and Site Layout Data Group. After processing is complete, the tester notes the time, then compares it against the time noted in the file *slla111111.ssf*, under the parameter MMSPTime. The tester also checks that only the appropriate data are copied to the GRF and that no warnings or errors remain after processing.

4.3.1.5 Results

The MMSP Module Execution Manager read and copied input first for the Land Application Unit Module, then for Farm Foodchain, Human Exposure, and Human Risk Modules. Results were written to the GRF directory. No warning or error files remained after processing. The MMSP Module Execution Manager produced GRFs for the three human receptors identified in the Site Layout Data Group, as well as Source, Farm Foodchain, and Site Layout Data Group GRF. Hand-timing results approximated MMSP Module Execution Manager timing results. Therefore, the MMSP Module Execution Manager passed this test case.

4.3.2 MMSP_02

4.3.2.1 Description and Rationale

As mentioned, the MMSP Module Execution Manager must follow cues in the Site Layout Data Group, as well as input from the user through the SUI, to ensure the computational modules are executed in the proper sequence. This test case evaluates the ability of the MMSP Module Execution Manager to model exposure from a landfill through the vadose zone to aquifer and farm foodchain, resulting in human exposure and risk (Figure 4.2).

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

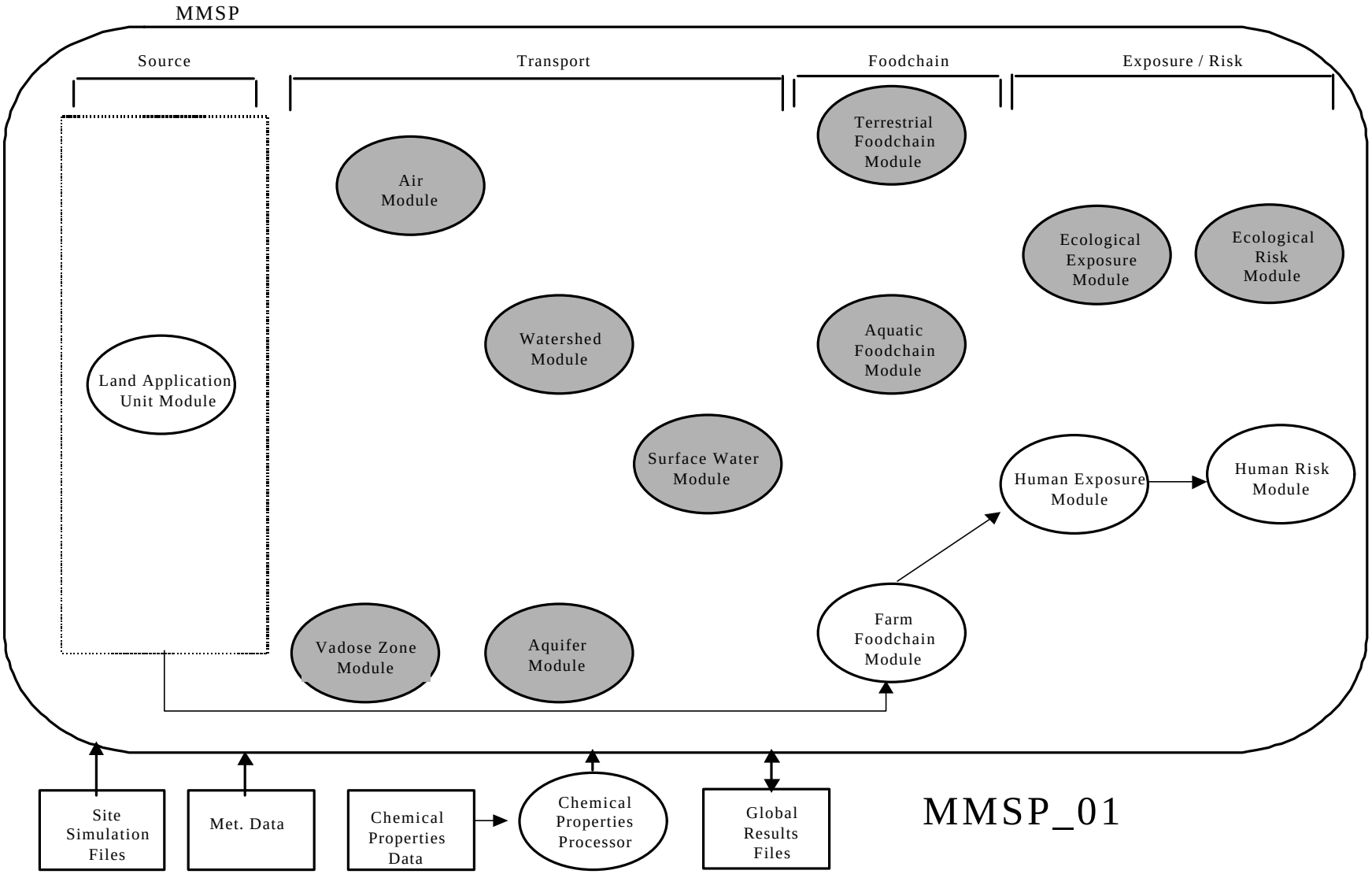


Figure 4.1 Sequence of Modules Tested in Test Case MMSP_01

4.3.2.2 Input Data

Data needed for input to the MMSP include the suite of SSF (with the Site Layout Data Group) and GRF. The command lines necessary to run this test case are in file *runMMSP-02.bat*. This batch file passes the following commands to the MMSP Module Execution Manager:

```
del grf\*.grf
MMSP ssf grf hd222222.ssf
```

See the description for Test Case MMSP_01 (Section 4.3.1.2) for an explanation of these commands.

In addition, this test uses the header file *hd222222.ssf* and Site Layout Data Group file *sllf222222.ssf*, which direct the MMSP Module Execution Manager to execute the computational modules in the sequence described in Section 4.3.2.3.

4.3.2.3 Expected Results

The MMSP Module Execution Manager runs all modules without errors or warnings. Computational modules in process are approximated by reading and copying their SSF input files.

The SSF for the Landfill Module is read first, followed by those for the Vadose Zone, Aquifer, Farm Foodchain, Human Exposure, and the Human Risk Modules. No other input files are used. Results are written to the GRF location specified by the MMSP Module Execution Manager (in this case, the GRF directory). In addition, no errors or warnings remain after processing. Hand-timing results approximate MMSP Module Execution Manager timing results within 2 seconds.

4.3.2.4 Procedure

The tester uses a stop watch or other clock to determine time in seconds. Using the Windows® Explorer, the tester double-left-mouse-clicks on the file *runMMSP-02.bat*, noting the time. The MMSP Module Execution Manager accesses the simulated SSF, including the appropriate header file and Site Layout Data Group. After processing is complete, the tester notes the time, then compares it against the time noted in the file *sllf222222.ssf*, under the parameter MMSPTime. The tester also checks that only the appropriate data are copied to the GRF and that no warnings or errors remain after processing.

4.3.2.5 Results

The MMSP Module Execution Manager read and copied input first for the Landfill Module, then for the Vadose Zone Module, Aquifer Module, Farm Foodchain Module, Human Exposure Module, and the Human Risk Module. Results were written to the GRF directory. No warning or error files remained after processing. The MMSP Module Execution Manager produced GRF for the three human receptors identified in the Site Layout Data Group, as well as the two farms, the aquifer, the landfill, and the source. Hand-timing results approximated MMSP Module Execution Manager timing results. Therefore, the MMSP Module Execution Manager passed this test case.

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

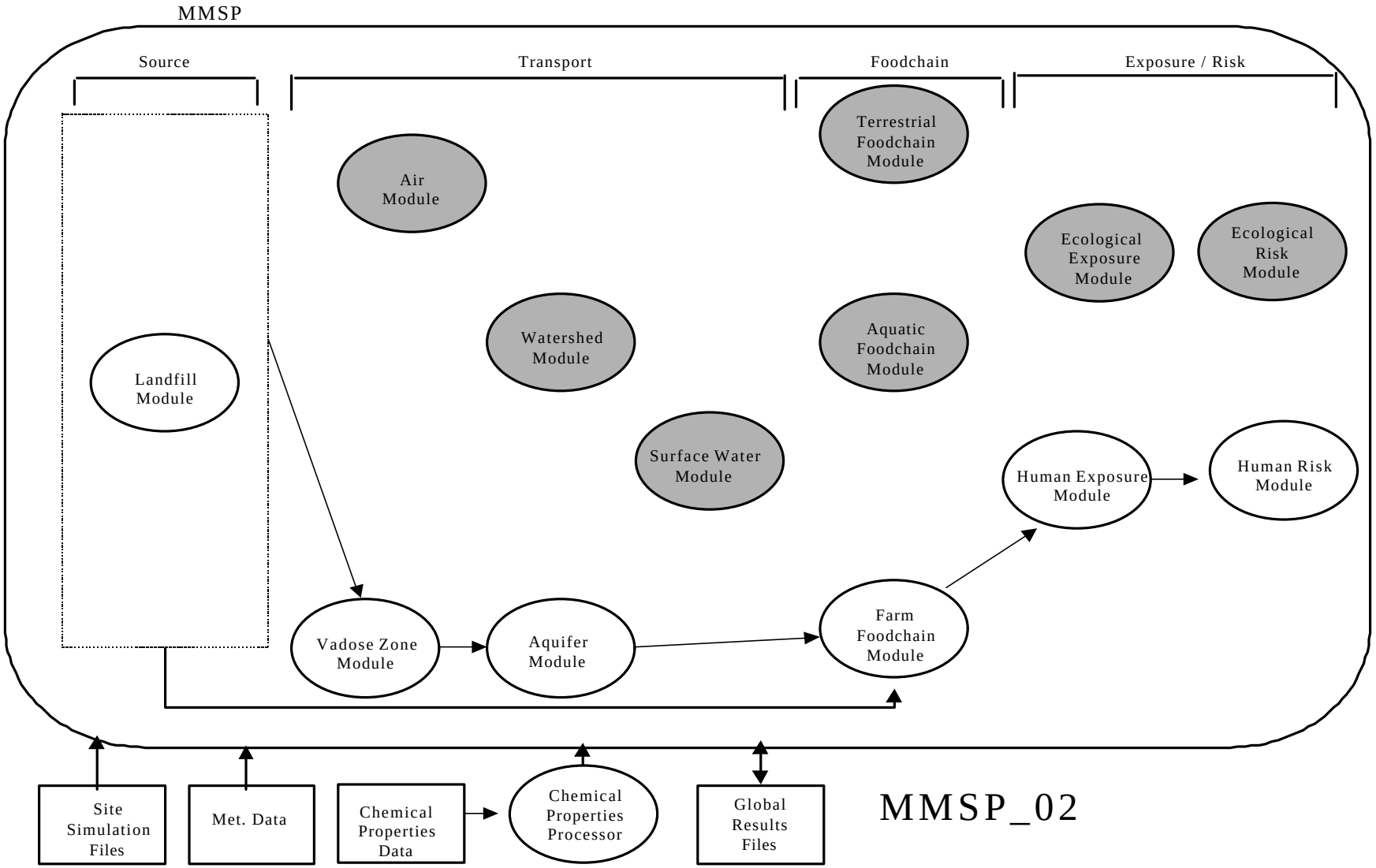


Figure 4.2 Sequence of Modules Tested in Test Case MMSP_02

4.3.3 MMSP_03

4.3.3.1 Description and Rationale

As mentioned, the MMSP must follow cues in the Site Layout Data Group, as well as input from the user through the SUI, to ensure the computational modules are executed in the proper sequence. This test case evaluates the ability of the MMSP Module Execution Manager to execute the Aerated Tank Module with a single primary pathway of contamination (air) resulting in human and ecological exposure and risk (Figure 4.3).

4.3.3.2 Input Data

Data needed for input to the MMSP include the suite of SSF (with the Site Layout Data Group) and GRF. The command lines necessary to run this test case are in the file *runMMSP-03.bat*. This batch file passes the following commands to the MMSP Module Execution Manager:

```
del grf\*.grf
MMSP ssf grf hd333333.ssf
```

See the description for Test Case MMSP_01 (Section 4.3.1.2) for an explanation of these commands.

In addition, this test uses the header file *hd333333.ssf* and Site Layout Data Group file *slat333333.ssf*, which directs the MMSP Module Execution Manager to execute the computational modules in the sequence described in Section 4.3.3.3.

4.3.3.3 Expected Results

The MMSP Module Execution Manager runs all modules without errors or warnings. Computational modules in process are approximated by reading and copying their SSF input files. The SSF for the Aerated Tank Module is read first, followed by those for the Air Module, Watershed Module, Surface Water Module, Aquatic Foodchain Module, Farm Foodchain Module, Terrestrial Foodchain Module, Ecological Exposure Module, Ecological Risk Module, Human Exposure Module, and the Human Risk Module. No other input files are used. Results are written to the GRF directory. In addition, no errors or warnings remain after processing. Hand-timing results approximate MMSP Module Execution Manager timing results within 2 seconds.

4.3.3.4 Procedure

The tester uses a stop watch or other clock to determine time in seconds. Using the Windows® Explorer, the tester double-left-mouse-clicks on the file *runMMSP-03.bat*, noting the time. The MMSP Module Execution Manager accesses the simulated SSF, including the appropriate header file and Site Layout Data Group. After processing is complete, the tester notes the time, then compares it against the time noted in the file *slat333333.ssf*, under the parameter MMSPTime. The tester also checks that only the appropriate data are copied to the GRF and that no warnings or errors remain after processing.

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

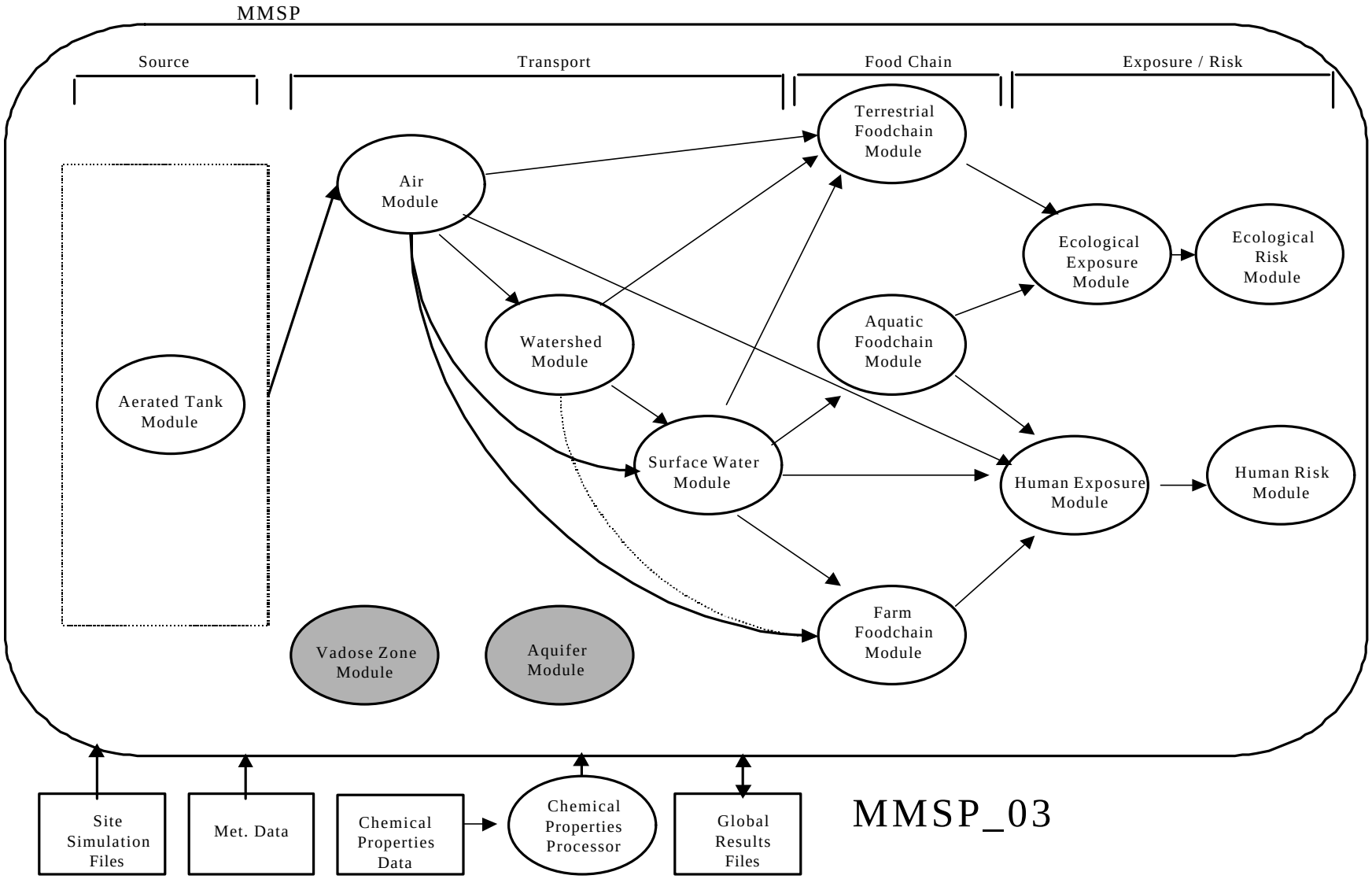


Figure 4.3 Sequence of Modules Tested in Test Case MMSP_03

4.3.3.5 Results

The MMSP Module Execution Manager read and copied input first for the Aerated Tank Module, then for the Air Module, Watershed Module, Surface Water Module, Aquatic Foodchain Module, Farm Foodchain Module, Terrestrial Foodchain Module, Ecological Exposure Module, Ecological Risk Module, Human Exposure Module, and the Human Risk Module. Results were written to the GRF directory. No warning or error files remained after processing. The MMSP Module Execution Manager produced GRF for the 3 human receptors identified in the Site Layout Data Group, as well as for the 3 fish, 10 air segments, 2 ecological receptors, 2 farms, 1 source, 3 rivers, 2 lakes, 2 game, and 3 watersheds. Hand-timing results approximated MMSP Module Execution Manager results. Therefore, the MMSP Module Execution Manager passed this test case.

4.3.4 MMSP_04

4.3.4.1 Description and Rationale

As mentioned, the MMSP must follow cues in the Site Layout Data Group, as well as input from the user through the SUI, to ensure the computational modules are executed in the proper sequence. This test case evaluates the ability of the MMSP Module Execution Manager to execute another of the more limited scenarios, the surface impoundment through two pathways (air and vadose zone) resulting in human and ecological exposure and risk (Figure 4.4).

4.3.4.2 Input Data

Data needed for input to the MMSP include the suite of SSF (with the Site Layout Data Group) and GRF. The commands necessary to run this test case are found in the file *runMMSP-04.bat*.

This batch file passes the following commands to the MMSP Module Execution Manager:

```
del grf\*.grf
MMSP ssf grf hd444444.ssf
```

See the description for Test Case MMSP_01 (Section 4.3.1.2) for an explanation of these commands.

In addition, this test uses the header file *hd444444.ssf* and Site Layout Data Group file *slsi444444.ssf*, which direct the MMSP Module Execution Manager to execute the computational modules in the sequence described in Section 4.3.4.3.

4.3.4.3 Expected Results

The MMSP Module Execution Manager runs all modules without errors or warnings. Computational modules in process are approximated by reading and copying their SSF input files. The SSF for the Surface Impoundment Module is read first, followed by the Vadose Zone Module, Aquifer Module, Air Module, Watershed Module, Surface Water Module, Aquatic Foodchain Module, Farm Foodchain Module, Terrestrial Foodchain Module, Ecological Exposure Module, Ecological Risk Module, Human

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

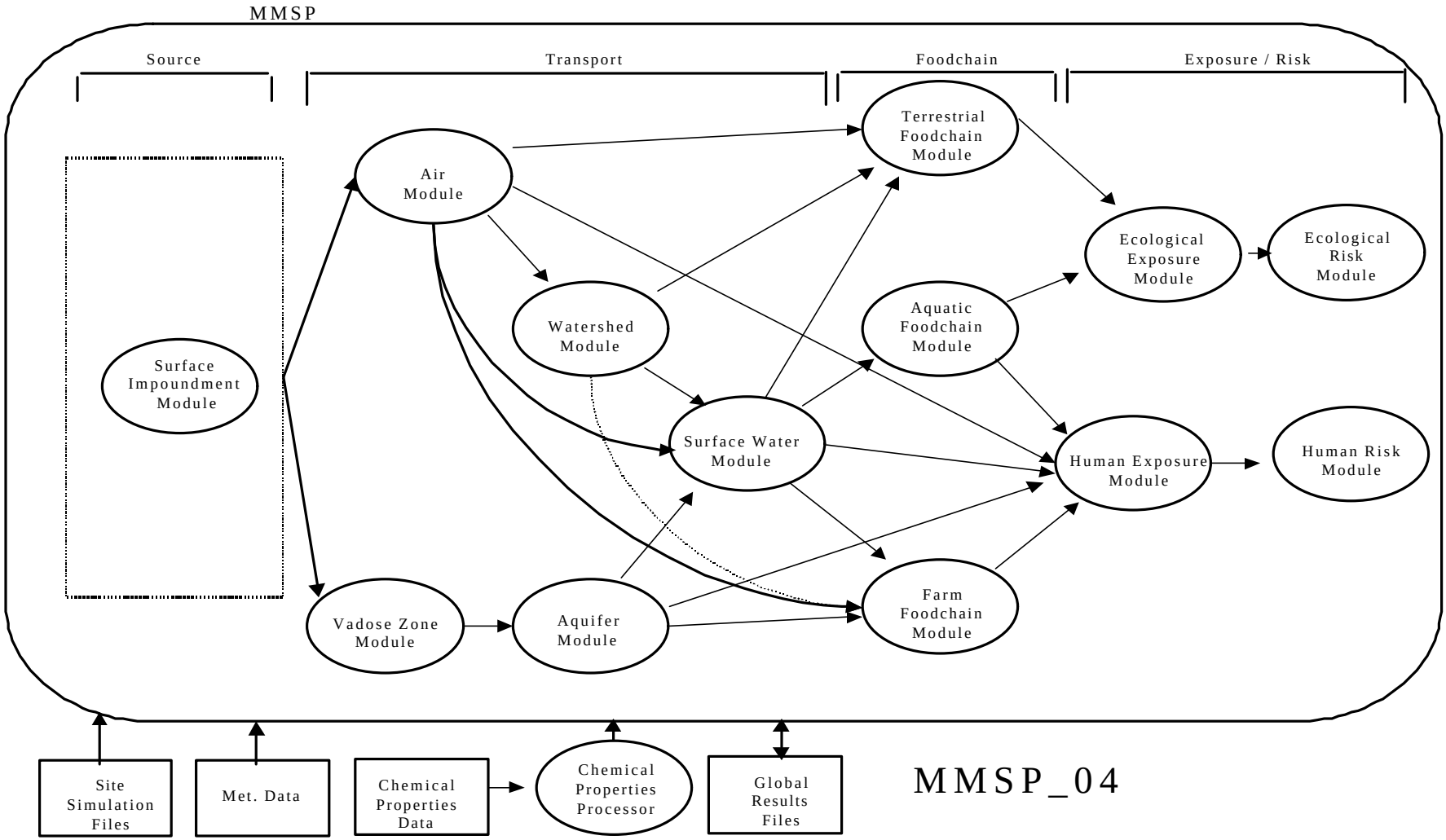


Figure 4.4 Sequence of Modules Tested in Test Case MMSP_04

Exposure Module, and Human Risk Module. No other input files are used. Results are written to the GRF directory. In addition, no errors or warnings remain after processing. Hand-timing results approximate MMSP Module Execution Manager timing results within 2 seconds.

4.3.4.4 Procedure

The tester uses a stop watch or other clock to determine time in seconds. Using the Windows® Explorer, the tester double-left-mouse-clicks on the file *runMMSP-04.bat*, noting the time. The MMSP Module Execution Manager accesses the simulated SSF, including the appropriate header file and Site Layout Data Group. After processing is complete, the tester notes the time, then compares it against the time noted in the file *slsi444444.ssf*, under the parameter MMSPTime. The tester also checks that only the appropriate data are copied to the GRF and that no warnings or errors remain after processing.

4.3.4.5 Results

The MMSP Module Execution Manager read and copied input first for the Surface Impoundment Module, then for the Vadose Zone Module, Aquifer Module, Air Module, Watershed Module, Surface Water Module, Aquatic Foodchain Module, Farm Foodchain Module, Terrestrial Foodchain Module, Ecological Exposure Module, Ecological Risk Module, Human Exposure Module, and Human Risk Module. Results were written to the GRF directory. No warning or error files remained after processing. The MMSP Module Execution Manager produced GRFs for the 3 human receptors identified in the Site Layout Data Group, as well as the 3 fish, 1 aquifer, 10 air segments, 2 ecological exposures, 2 farms, 1 site layout, 1 source, 3 rivers, 2 lakes, 2 game, 2 vadose zones, and 3 watersheds. Hand-timing results approximated MMSP Module Execution Manager timing results. Therefore, the MMSP Module Execution Manager passed this test case.

4.3.5 MMSP_05

4.3.5.1 Description and Rationale

As mentioned, the MMSP must follow cues in the Site Layout Data Group, as well as input from the user through the SUI, to ensure the computational modules are executed in the proper sequence. This test case evaluates the ability of the MMSP Module Execution Manager to execute a waste pile source through multiple pathways (air, vadose zone, watershed, surface water, and farm foodchain), resulting in human and ecological exposure and risk (Figure 4.5).

4.3.5.2 Input Data

Data needed for input to the MMSP include the suite of SSF (with the Site Layout Data Group) and GRF. The command lines necessary to run this test case are found in the file *runMMSP-05.bat*. This batch file passes the following commands to the MMSP Module Execution Manager:

```
del grf\*.grf
MMSP ssf grf hd555555.ssf
```

See the description for Test Case MMSP_01 (Section 4.3.1.2) for an explanation of the remaining commands.

In addition, this test uses the header file hd555555.ssf and Site Layout Data Group file *slwp555555.ssf*, which direct the MMSP Module Execution Manager to execute the computational modules in the sequence described in Section 4.3.5.3.

US EPA ARCHIVE DOCUMENT

4.3.5.3 Expected Results

The MMSP Module Execution Manager runs all modules without errors or warnings. Computational modules in process are approximated by reading and copying their SSF input files. The SSF for the Waste Pile Module is read first, followed by the Vadose Zone Module, Aquifer Module, Air Module, Watershed Module, Surface Water Module, Aquatic Foodchain Module, Farm Foodchain Module, Terrestrial Foodchain Module, Ecological Exposure, Ecological Risk Module, Human Exposure Module, and Human Risk Module. No other input files are used. Results are written to the GRF directory. In addition, no errors or warnings remain after processing. Hand-timing results approximate MMSP Module Execution Manager timing results within 2 seconds.

4.3.5.4 Procedure

The tester uses a stop watch or other clock to determine time in seconds. Using the Windows® Explorer, the tester double-left-mouse-clicks on the file *runMMSP-04.bat*, noting the time. The MMSP Module Execution Manager accesses the simulated SSF, including the appropriate header file and Site Layout Data Group. After processing is complete, the tester notes the time, then compares it against the time noted in the file *slwp555555.ssf* under the parameter MMSPTime. The tester also checks that only the appropriate data are copied to the GRF and that no warnings or errors remain after processing.

4.3.5.5 Results

The MMSP Module Execution Manager read and copied input first for the Waste Pile Module, then for the Vadose Zone Module, Aquifer Module, Air Module, Watershed Module, Surface Water Module, Aquatic Foodchain Module, Farm Foodchain Module, Terrestrial Foodchain Module, Ecological Exposure, Ecological Risk Module, Human Exposure Module, and Human Risk Module. No warning or error files remained after processing. The MMSP Module Execution Manager produced GRF for the 3 human receptors identified in the Site Layout Data Group, as well as an aquifer, 10 air segments, 3 site layouts, source, 3 rivers, 2 vadose zones, and 3 watersheds. Hand-timing results approximated MMSP Module Execution Manager timing results. Therefore, the MMSP Module Execution Manager passed this test case.

4.3.6 MMSP_06

4.3.6.1 Description and Rationale

As mentioned, the MMSP must follow cues in the Site Layout Data Group, as well as input from the user through the SUI, to ensure the computational modules are executed in the proper sequence. This test case evaluates the ability of the MMSP Module Execution Manager to execute the most complex of the scenarios, using contamination from a waste pile to all pathways, resulting in human and ecological exposure and risk (Figure 4.6).

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

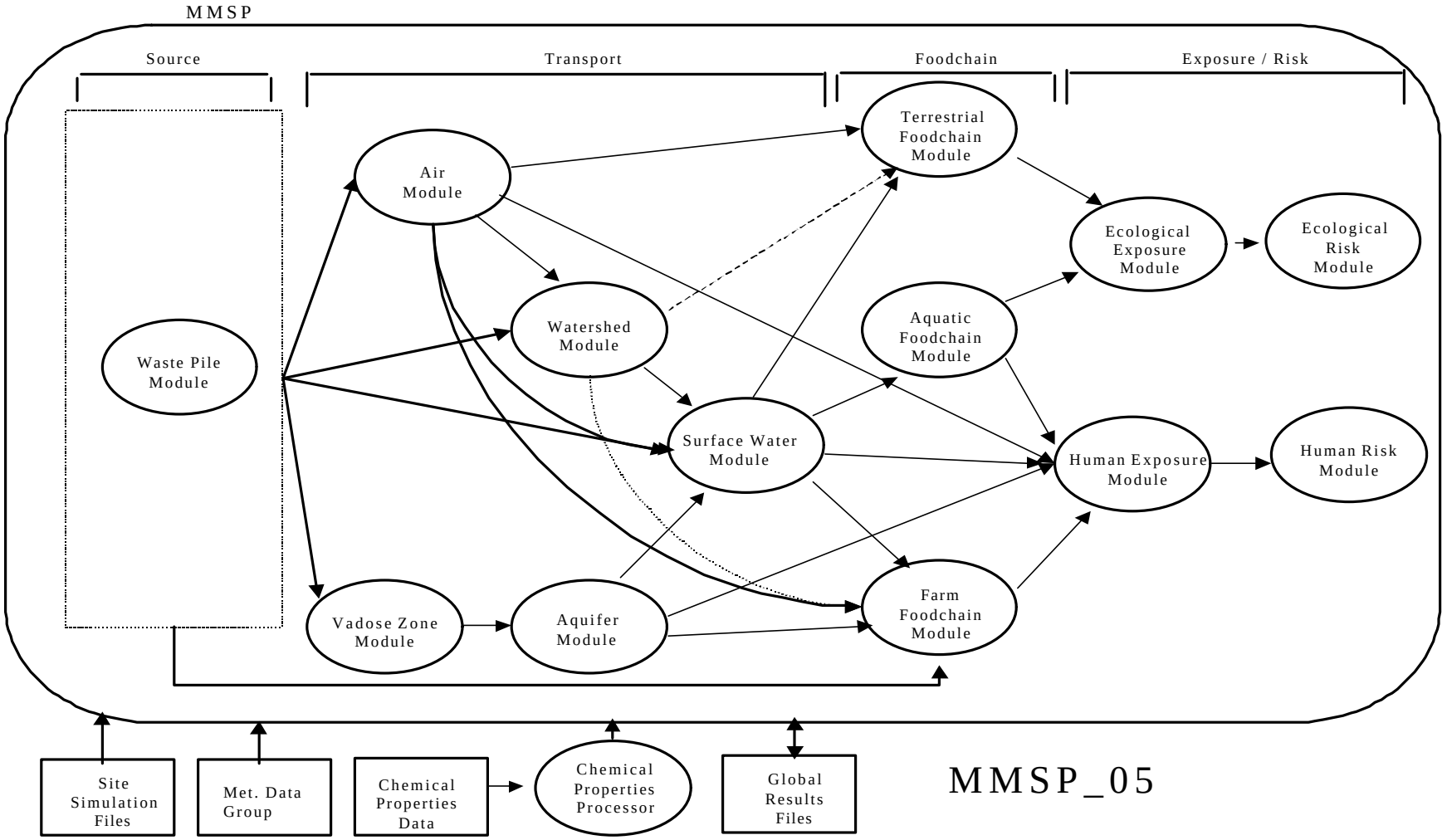


Figure 4.5 Sequence of Modules Tested in Test Case MMSP_05

4.3.6.2 Input Data

Data needed for input to the MMSP include the suite of SSF (with the Site Layout Data Group) and GRF. The command lines necessary to run this test case are found in the file *runMMSP-06.bat*. This batch file passes the following commands to the MMSP Module Execution Manager:

del grf*.grf
MMSP ssf grf hd666666.ssf

See the description for Test Case MMSP_01 (Section 4.3.1.2) for an explanation of the remaining commands.

In addition, this test uses the header file *hd666666*.ssf and Site Layout Data Group file *slwp666666.ssf*, which direct the MMSP Module Execution Manager to execute the computational modules in the sequence described in Section 4.3.6.3.

4.3.6.3 Expected Results

The MMSP Module Execution Manager runs all modules without errors or warnings. Computational modules in process are approximated by reading and copying their SSF input files. The SSF for the Waste Pile Module is read first, followed by those for the Vadose Zone Module, Aquifer Module, Air Module, Watershed Module, Surface Water Module, Aquatic Foodchain Module, Farm Foodchain Module, Terrestrial Foodchain Module, Ecological Exposure and Risk Module, Human Exposure Module, and Human Risk Module. Results are written to the GRF directory. In addition, no errors or warnings remain after processing. Hand-timing results approximate MMSP Module Execution Manager timing results within 2 seconds.

4.3.6.4 Procedure

The tester uses a stop watch or other clock to determine time in seconds. Using the Windows® Explorer, the tester double-left-mouse-clicks on the file *runMMSP-04.bat*, noting the time. The MMSP Module Execution Manager accesses the simulated SSF, including the appropriate header file and Site Layout Data Group. After processing is complete, the tester notes the time, then compares it against the time noted in the file *slsi444444.ssf* under the parameter MMSPTime. The tester also checks that only the appropriate data are copied to the GRF and that no warnings or errors remain after processing.

4.3.6.5 Results

The MMSP Module Execution Manager read and copied input first for the Waste Pile Module, then for the Vadose Zone Module, Aquifer Module, Air Module, Watershed Module, Surface Water Module, Aquatic Foodchain Module, Farm Foodchain Module, Terrestrial Foodchain Module, Ecological Exposure Module, Ecological Risk Module, Human Exposure Module, and the Human Risk Module. No warning or error files remained after processing. The MMSP Module Execution Manager produced GRF for the 3 human receptors identified in the Site Layout Data Group, as well as for the 3 fish, 1 aquifer, 10 air segments, 2 ecological exposures, 2 farms, 1 site layout, 1 source, 3 rivers, 2 lakes, 2 game, 2 vadose zones, and 3 watersheds. Hand-timing results approximated MMSP Module Execution Manager timing results. Therefore, the MMSP Module Execution Manager passed this test case.

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

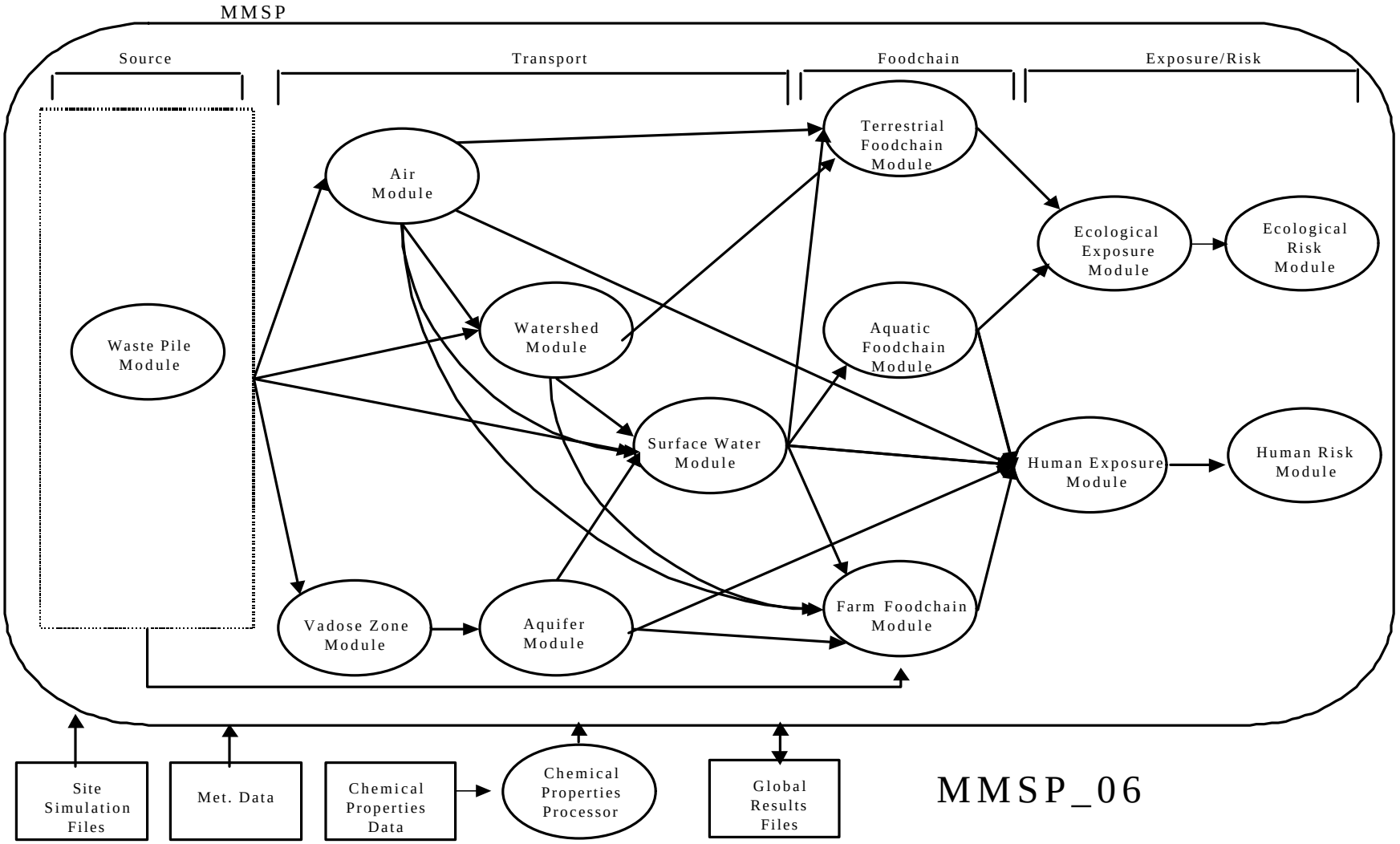


Figure 4.6 Sequence of Modules Tested in Test Case MMSP_06

4.3.7 MMSP_07

4.3.7.1 Description and Rationale

The FRAMES-HWIR Technology Software System serves several needs for the EPA. To support these needs, and to provide a flexible system for the computational modules, the MMSP Module Execution Manager provides for varying degrees of data storage (either minimal data in the GRF directory or all data in the permanent directory). Test Cases MMSP_01 through MMSP_06 evaluate the ability of the Module Execution Manager to save minimal data to the GRF. This test case evaluates the ability of the MMSP Module Execution Manager to save all data to permanent space, as directed by the user.

4.3.7.2 Input Data

Input data for this test case include the header file *hd777777.ssf*, the Site Layout Data Group file *slwp777777.ssf*, and other SSF. The Site Layout Data Group instructs the MMSP Module Execution Manager to run one source type (waste pile) through all transport pathways to result in both human and ecological exposure and risk. The command lines necessary to run this test case are found in the file *runMMSP-07.bat*. This batch file passes the following commands to the MMSP Module Execution Manager:

```
del grf\*.grf
MMSP ssf grf hd777777.ssf
```

See the description for Test Case MMSP_01 (Subsection 4.3.1.2) for an explanation of these commands.

4.3.7.3 Expected Results

The MMSP Module Execution Manager executes all computational modules and saves to the file folder *permanent* all data located in the MMSP directory. Minimal data are also saved to the GRF. In addition, no warnings or errors remain after processing.

4.3.7.4 Procedures

Using the Windows Explorer, the tester double-left-mouse-clicks on the file *runMMSP-07.bat*. The MMSP Module Execution Manager accesses the simulated SSF, including the appropriate Site Layout Data Group. After processing is complete, the tester accesses the MMSP directory and ensures that all data are saved to permanent space.

4.3.7.5 Results

The MMSP produced the expected GRF in the GRF Directory for all pathways. It also produced GRF for all pathways in the Permanent directory. No warning or error files remained after processing. Therefore, the MMSP passed this test case.

US EPA ARCHIVE DOCUMENT

4.3.8 MMSP_08

4.3.8.1 Description and Rationale

So the user can respond appropriately, the MMSP must pass information regarding its own errors to the GRF. Errors are defined as critical difficulties that result in abnormal processor termination. The information must contain an error description that allows the user or processor developer to correct the problem. The main processor-specific error for the MMSP relates to its ability to recognize inappropriate arguments it receives. This test case evaluates the ability of the MMSP Module Execution Manager to recognize and pass this error information.

4.3.8.2 Input Data

Input data for this test case include the command lines from Test Case MMSP_07. No matching header file or SSF are provided, making it impossible for the MMSP to function.

4.3.8.3 Expected Results

When attempting to process, the MMSP Module Execution Manager returns an error message to the GRF, specifying the data file cannot be found. The MMSP Module Execution Manager also terminates processing.

4.3.8.4 Procedures

Using the Windows® Explorer, the tester double-left-mouse-clicks on the file *runMMSP-08.bat*. The MMSP Module Execution Manager attempts to instruct the computational modules to execute. Before processing, the MMSP Module Execution Manager writes an error message to the GRF directory error file indicating the file could not be found and terminates. When processing terminates, the tester compares the message in the error file of the GRF directory with the expected message.

4.3.8.5 Results

After less than 3 seconds of processing, the MMSP terminated. It produced the following error message in a file in the GRF directory:

"Failed to call CloseGroups",
"Data group hd888888.ssf
Can not find/open data group",

Therefore, the MMSP passed this test case.

4.4 Verification Testing

In addition to tests conducted under unit testing, the MMSP underwent two tests, chosen by the EPA, for verification testing. The following information was supplied to the EPA, with additional data from the rest of this report, to document this testing.

US EPA ARCHIVE DOCUMENT

To perform verification testing, a dummy module batch file was used. This batch file wrote the MS-DOS environment variable Arguments and module name being executed to an output file. The sequence of module execution was checked against the sequence determined by the module's data needs.

- Step 1: Run WP0223504 (the site identification number for verification test 1) through the MMSP with the dummy module and check sequence--No errors found.
- Step 2: Run LA1632106 (the site identification number for verification test 2) through the MMSP with the dummy module and check sequence--No errors found.

Both of these test cases used full pathways (all modules). Because no errors were found, the MMSP passed verification testing.

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

5.0 Quality Assurance Program

The MMSP Module Execution Manager was developed under a quality assurance program documented in Gelston et al. (1998). The computational modules were developed under quality assurance programs that met the expectations in *Documentation for the FRAMES-HWIR Technology Software System, Volume 9: Software Development and Testing Strategies* (see Section 6.0). The following subsections summarize those programs.

5.1 Quality Assurance Program for the Module Execution Manager

Quality is defined as the ability of the software to meet client needs. Meeting client needs starts with a shared understanding of how the software must perform and continues throughout the software life cycle of design, development, testing, and implementation through attention to details.

Figure 5.1 outlines the software development process used for the MMSP Module Execution Manager, highlighting the quality check points. (Note the MMSP Module Execution Manager activities flow down the left side of the figure, because it is software developed for the first time, as opposed to a modification to existing software.) The process shown is designed for compatibility with similar processes used by other government agencies. For example, this quality process compares favorably with that in the EPA Directive 2182*, System Design and Development Guidance* (EPA 1997). It also compares favorably with the Office of Civilian Radioactive Waste Management's *Quality Assurance Requirements and Description, Supplement I, Software* (OCRWM 1995). Activities roughly equivalent across these processes are shown in Table 5.1.

Development of the MMSP Module Execution Manager included the implementation of a quality assurance check list (see Figure 5.2). Understanding of this checklist by all team members resulted in the shared understanding of component requirements and design necessary to ensure quality. Completion of this checklist verified that all documentation was complete for transfer of the software to client use.

5.2 Quality Assurance Expectations for the Modules

To ensure that all these components of the FRAMES-HWIR Technology Software System interact appropriately when placed into the software system, EPA requested that module and processor developers meet a set of expectations in the areas of quality assurance and testing.

In the area of quality assurance, module and processor developers were expected to:

- use an appropriate approach to quality assurance and documentation
- work with related-media modelers to ensure consistency of assumptions/data transfer between media (for example, a modeler creating a vadose zone module might need to ensure consistency with a modeler creating a aquifer module that would use vadose zone module results as input)
- provide documentation of the module and processor, including user's guidance, mathematical formulations, and documentation of requirements, design/specifications, and testing.

US EPA ARCHIVE DOCUMENT

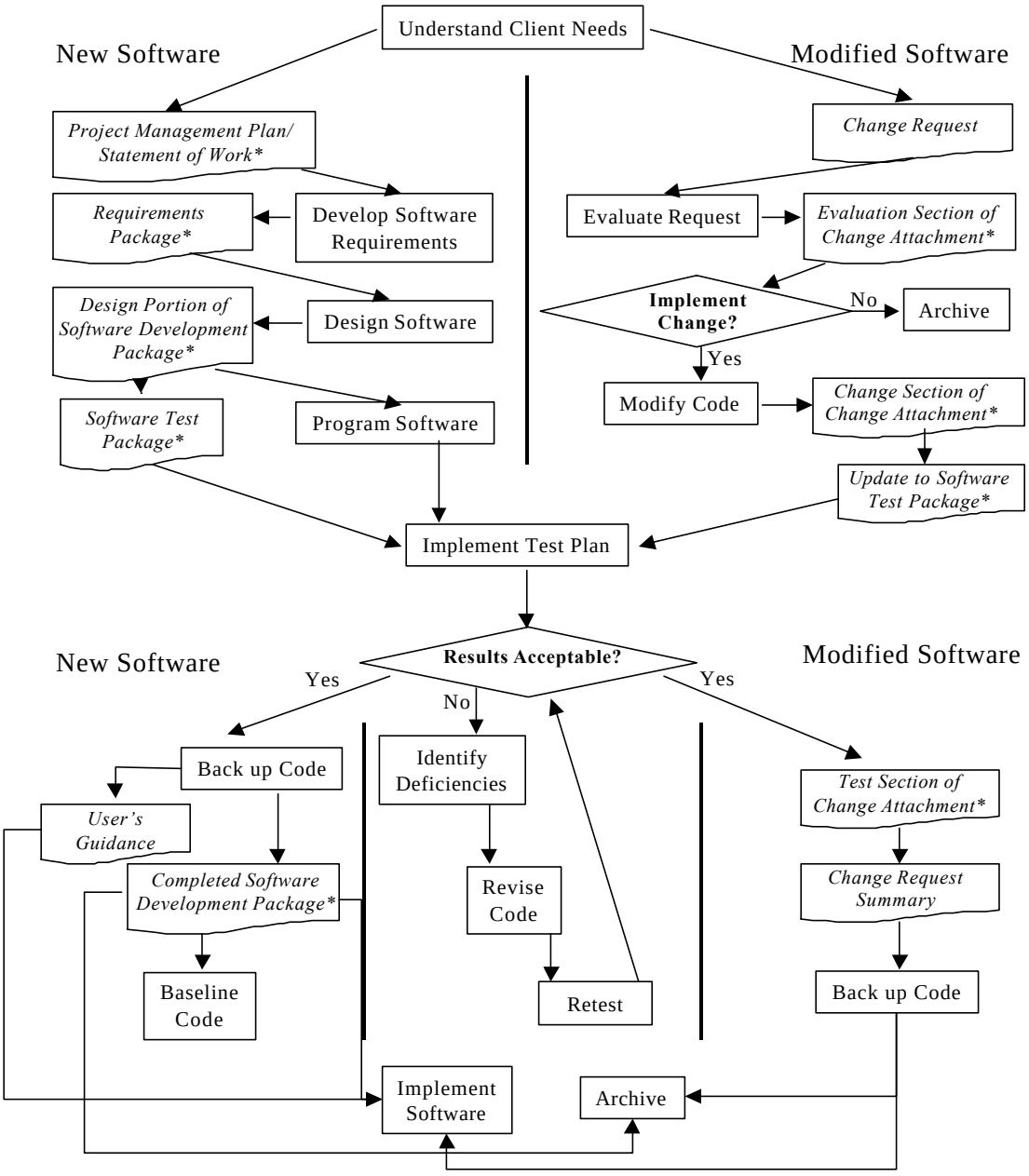


Figure 5.1 Ensuring Quality in the Environmental Software Development Process (* indicates quality review stage; box with wavy bottom line and italics font indicates a document rather than an activity)

Table 5.1 Relationship of PNNL Environmental Software Development Process to Quality Assurance Requirements (OCRWM 1995, EPA 1997)

OCRWM Quality Assurance Requirement[a]	EPA Essential Element of Information[b]	Environmental Software Process Equivalent (Section)
	4—System Implementation Plan	Project Management Plan or Statement of Work
I.2.5A Functional Requirements Information Documentation; I.2.5C Requirements and Design Documentation	5—System Detailed Requirements Document	Requirements Package
I.2.1 Software Life Cycles, Baselines (see their Appendix C), and Controls	6—Software Management Plan	Project Management Plan or Statement of Work and Gelston et al. (1998)
I.2.2 Software Verification[c] and Software Validation; I.2.4 Software Validation[d]	7—Software Test and Acceptance Plan	Software Test Package
I.2.3 Software Verification; I.2.5C Requirements and Design Information Documentation	8—Software Design Document	Design Portion of Software Development Package
I.2.6A Configuration Identification		Completed Software Development Package
I.2.6B Configuration Control; I.2.6C Configuration Status; I.2.7 Defect Reporting and Resolution[e]	9—Software Maintenance Document	Modification Documentation
	10—Software Operations Document	User's Guidance and Training
I.2.5B User Information Documentation	11—Software User's Reference Guide	User's Guidance and Training
	12—System Integration Test Reports	Software Test Package

(a) Note that OCRWM requirement I.2.8, Control of the Use of Software, is the responsibility of the OCRWM-related client.
(b) Elements 1 through 3 are generally completed by clients in the U.S. Environmental Protection Agency before contract initiation with the project team.
(c) Verification includes informal code testing by software engineers (see their Appendix C) to ensure that code functions as required.
(d) Validation includes testing by those other than the software engineers who developed the code to provide an independent confirmation that software functions as required.
(e) Note that some changes requested by clients may not be made in the software unless funding has been allocated for such modifications.

US EPA ARCHIVE DOCUMENT

A. General Requirements Analysis
--Documented in
_____Statement of Work (stored in project file; see Gene Whelan, Gariann Gelston, or current Integration Leader)
--Contains information on (all of the following)
_____problem description
_____deliverables
_____project team
_____capabilities to be used
_____restrictions
_____difficulties envisioned
_____compatibilities with existing software/hardware
_____scope of the project

B. Specific Requirements Analysis
--Documented in
_____requirements section of documentation (PNNL-11914, Volume 6, Section 2.0)
--Contains information on (all of the following)
_____purpose of the software
_____structure of the software
_____hardware and software requirements
_____input and output requirements
_____scientific basis
_____assumptions
_____limitations
_____post-October 31 requirements

C. Design Documentation
--Documented in
_____design portion of documentation (PNNL-11914, Volume 6, Section 3.0)
_____team task plans/Project Management Plan (stored in project file; see Gene Whelan, Gariann Gelston, or current Integration Leader)
--Contains information on (all of the following)
_____code type and description
_____development team members
_____specifications
_____logic diagrams
_____"help" descriptions
_____methods to ensure consistency in components
_____mathematical formulations
_____need for pre-postprocessors
_____post-October 31 design elements

D. Development Documentation
--Documented in
_____Specifications Document (PNNL-11914, Volume 8)
_____Quality Assurance Archive (see Gariann Gelston or current Integration Leader)
--Contains information on (all of the following)
_____baseline hard copy of the source code
_____diskette copy
_____name of computer language(s) used

E. Testing Documentation
--Documented in
_____test plan that meets quality assurance requirements (PNNL-11914, Volume 6, Section 4.0)
--Contains information on (all of the following)
_____description of software
_____testing scope
_____relationship between test cases and requirements
_____test activity description
_____hardware and software needed to implement plan
_____test case specifications
_____expected results

Figure 5.2 Quality Assurance Implementation Checklist for the Module Execution Manager

US EPA ARCHIVE DOCUMENT

F. User's Guidance

--Documented in

_____hardcopy printout of user's guidance for system (PNNL-11914, Volume 11)

--Contains information on (all of the following)

_____description of software

_____description of use of user interface

_____mathematical formulations

_____example problems

_____explanation of modules included

G. General Quality Assurance Documentation

--Documented in

_____Quality Assurance Program Document (PNNL-11880)

_____Quality Assurance Software-Specific Checklist (PNNL-11914, Volume 6, Section 5.0)

--Contains information on (all of the following)

_____purpose of quality assurance program

_____client-specified activities

_____activities required to ensure quality in software

H. Quality Assurance Archive

--Documented in

_____hardcopy files (see Gariann Gelston or current Integration Leader)

_____back up disk files in multiple storage locations (see Gariann Gelston or current Integration Leader)

--Contains information on (all of the following)

_____all quality assurance documentation

_____client correspondence regarding software

_____modifications made to baselined software

_____disk copy back ups

_____reproducibility of code (check code for comments)

Completed by _____ Date _____

Approved by
System/Module Manager _____ Date _____

Figure 5.2 Quality Implementation Checklist (contd)

At a later date, further information on the MMSP computational modules will be published by the EPA as separate reports that constitute appendixes to this report. Figure 5.3 shows the internal testing process followed by the modules.

In the area of testing, module and processor developers were expected to

- develop a test plan
- ensure that unit testing of programs was thorough and well documented
- ensure that integration testing of the pieces composing their processor was thorough and well documented

US EPA ARCHIVE DOCUMENT

- ensure that processors communicate with the wider system through the shared routines, such as the HWIRIO.DLL and the spatial and temporal integrator DLL and document such system tests

- revise requirements documentation, design/specifications documentation, and program code implementation as needed to resolve issues found during testing

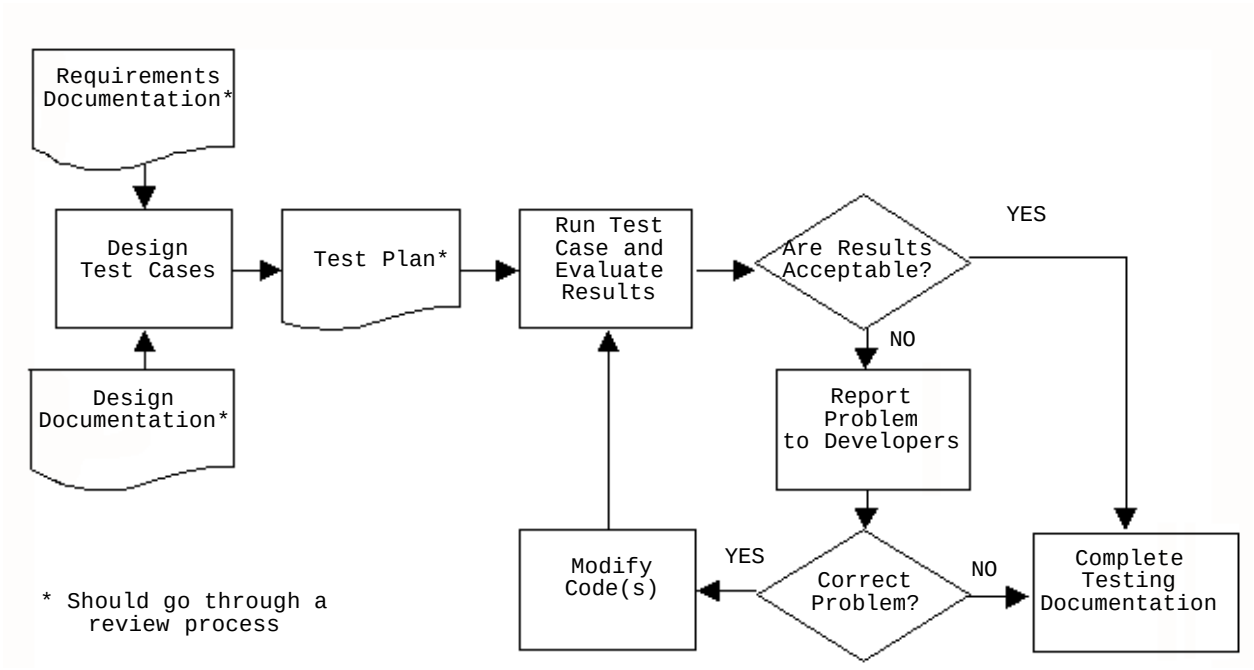
- document computational module and processor limitations, addressing those limitations as needed in coding to ensure the program functions as intended and eliminating those limitations that inhibit the required functionality of the software

- provide documentation from internal testers to support the conclusion that software meets its requirements follow the process shown in Figure 5.2 for internal testing (from *Documentation for the FRAMES-HWIR Technology Software System, Volume 9: Software Development and Testing Strategies*, see Section 6.0).

- provide the computational module or processor executable, the source code, the test plan, and test results to an external independent test group for verification

- work iteratively with the external independent test group as needed to resolve issues.

Figure 5.3 Internal Testing Process Followed by Modules



US EPA ARCHIVE DOCUMENT

6.0 References

Documentation for the FRAMES-HWIR Technology Software System

Volume 1: Overview of the FRAMES-HWIR Technology Software System. 1998. PNNL-11914, Vol. 1, Pacific Northwest National Laboratory, Richland, Washington.

Volume 2: System User Interface Documentation. 1998. PNNL-11914, Vol. 2, Pacific Northwest National Laboratory, Richland, Washington.

Volume 3: Distribution Statistics Processor Documentation. 1998. TetraTech, Lafayette, California.

Volume 4: Site Definition Processor Documentation. 1998. PNNL-11914, Vol. 4, Pacific Northwest National Laboratory, Richland, Washington.

Volume 5: Computational Optimization Processor Documentation. 1998. TetraTech, Lafayette, California.

Volume 6: Multimedia Multipathway Simulation Processor Documentation. 1998. PNNL-11914, Vol. 6, Pacific Northwest National Laboratory, Richland, Washington.

Volume 7: Exit Level Processor Documentation. 1998. PNNL-11914, Vol. 7, Pacific Northwest National Laboratory, Richland, Washington.

Volume 8: Specifications. 1998. PNNL-11914, Vol. 8, Pacific Northwest National Laboratory, Richland, Washington.

Volume 9: Software Development and Testing Strategies. 1998. PNNL-11914, Vol. 9, Pacific Northwest National Laboratory, Richland, Washington.

Volume 10: Software Development Kit. 1998. PNNL-11914, Vol. 10, Pacific Northwest National Laboratory, Richland, Washington.

Volume 11: User's Guidance. 1998. PNNL-11914, Vol. 11, Pacific Northwest National Laboratory, Richland, Washington.

Volume 12: Dictionary. 1998. PNNL-11914, Vol. 12, Pacific Northwest National Laboratory, Richland, Washington.

Volume 13: Chemical Properties Processor Documentation. 1998. PNNL-11914, Vol. 13, Pacific Northwest National Laboratory, Richland, Washington.

Volume 14: Site Layout Processor Documentation. 1998. PNNL-11914, Vol. 14, Pacific Northwest National Laboratory, Richland, Washington.

Volume 15: Risk Visualization Tool Documentation. 1998. PNNL-11914, Vol. 15, Pacific Northwest National Laboratory, Richland, Washington.

US EPA ARCHIVE DOCUMENT

Quality Assurance Program Document

Gelston, G. M., R. E. Lundgren, J. P. McDonald, and B. L. Hoopes. 1998. *An Approach to Ensuring Quality in Environmental Software*. PNNL-11880, Pacific Northwest National Laboratory, Richland, Washington.

Additional References

Office of Civilian Radioactive Waste Management (OCRWM). 1995. *Quality Assurance Requirements and Description, Supplement I, Software*. U.S. Department of Energy, Washington, D.C.

U.S. Environmental Protection Agency (EPA). 1997. *System Design and Development Guidance*. EPA Directive Number 2182, Washington, D.C.

Marin, C., and Z. Saleem. 1997. *A Preliminary Framework for Finite-Source Multimedia, Multipathway and Multireceptor Risk Assessment (3MRA)*. Draft, October 1997, U.S. Environmental Protection Agency, Office of Solid Waste, Washington, D.C.

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

Appendix A

Additional Testing Information

Appendix A
Additional Testing Information

The Multimedia Multipathway Simulation Processor (MMSP) Module Execution Manager was tested using the protocols described in Section 4.0 of this document. This appendix describes the specifications of the Site Layout Data Group that is critical to testing the MMSP and provides a procedure for setting up test cases for unit testing.

A.1 Specifications of the Site Layout Data Group

The MMSP Module Execution Manager receives its cues about which computational modules to execute in what sequence from the Site Layout Data Group, which is populated by information provided by the user through the System User Interface (header file) and data from the Site Definition Processor. Because these components were not available for unit testing, a generic Site Layout Data Group was developed for unit testing of the MMSP. Exact values are available upon request. The following listing provides the specifications for manually creating a Site Layout Data Group.

Type GeoReference Point
NS(Float) m ; Northing in UTM
EW(Float) m ; Easting in UTM
Elev(Float) m ; Elevation

Type Source Layout
NumSrc(Integer) ; Number of sources at the site
SrcId(String)(NumSrc) ; Environmental setting identification for the aquifer
SrcName(String)(NumSrc) ; Identification of the site and sources
SrcType(String)(NumSrc) ; One of five types: *"LAU","LF","WP","AT","SI"*
SrcLoc(GeoReferencePoint)(NumSrc) ; see GeoReference Point information for description
SrcArea(Float)(NumSrc) ; Area of source, in square meters
SrcOvl(Logical)(NumSrc) ; Is overland flow to be computed from a local watershed to stream or lake
SrcLeach(Logical)(NumSrc) ; Is infiltration to vadose or groundwater computed
SrcAtm(Logical)(NumSrc) ; Is an air release computed

Type Vadose Zone Layout
NumVad(Integer) ; Number of vadose zones
VadId(String)(NumVad) ; Environmental setting identification for the aquifer
VadName(String)(NumVad) ; Name of the vadose zone
VadLoc(GeoReferencePoint)(NumVad) ; Location of vadose zone
VadNumSrc(Integer)(NumVad) ; Number of sources impacting this vadose zone
VadSrcIndex(Integer)(NumVad,VadNumSrc) ; Index in source
VadSrcFrac(Float)(NumVad,VadNumSrc) ; Fraction of source that impacts vadose zone
VadThick(Float)(NumVad) ;Vadose zone thickness in meters

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

Type Aquifer Layout

NumAqu(Integer) ; Number of aquifers
AquId(String)(NumAqu) ; Environmental setting identification for the aquifer
AquName(String)(NumAqu) ; Name of the aquifer
AquNumVad (Integer)(NumAqu) ; Number of vadose zones impacting the aquifer
AquVadIndex(Integer)
(NumAqu,Aqu NumVad) ; Index of vadose zones to aquifer
AquVadFrac(Float)(NumAqu,AquNumVad) ; Fraction of vadose zone that impacts aquifer
AquNumSrc(Integer)(NumAqu) ; Number of sources impacting the aquifer
AquSrcIndex(Integer)(NumAqu,AquNumSrc) ; Index to sources to aquifer
AquSrcFrac(Float)(NumAqu,AquNumSrc) ; Fraction of source that impacts aquifer
AquDir(Float)(NumAqu) ; Groundwater flow direction in degrees from North
AquNumWell(Integer)(NumAqu) ; Number of wells in an aquifer
AquWellType(String)(NumAqu,NumWell) ; “Drinking Water”,” Irrigation”,” DW&I”
AquWellLoc(GeoReferencePoint) ; Location of wells in aquifer

Type Surface Water Layout

NumLake(Integer) ; Number of lakes
LakeId ; Environmental setting identification of lake
LakeName(String)(NumLake) ; Name of lake
LakeLoc(GeoreferencePoint)(NumLake) ; Location of lake
LakeLocElev(Integer)(NumLake) ; Elevation of lake
LakeNumComp(Integer)(NumLake) ; Number of lake compartments
LakeCompLoc(GeoReferencePoint) ; Location of lake compartment
LakeCompNumAir(Integer)
(NumLake,LakeNumComp) ; Number of angular segments that impact lake compartment
LakeCompAirIndex(Integer)
(NumLake,LakeNumComp,
LakeCompNumAir) ; Index of air angular segment
LakeCompAirFrac(Float)(NumLake,
LakeNumComp,LakeCompNumAir) ; Fraction of lake area impacted by the angular segment
LakeCompNumWS(Integer)
(NumLake,LakeNumComp) ; Number of watersheds that impact lake compartment
LakeCompWSIndex(Integer)(NumLake,
LakeNumComp,LakeCompNumWS) ; Index of watershed that impacts lake compartment
LakeCompWSFrac(Float)(NumLake,
LakeNumComp,LakeCompNumWS) ; Fraction of lake area impacted by the watershed
LakeCompNumStrm(Integer)
(NumLake,LakeNumComp) ; Number of streams that impact lake compartment
LakeCompStrmIndex(Integer)(NumLake,
LakeNumComp,LakeCompNumStrm) ; Index of stream that impacts lake compartment
LakeCompStrmFrac(Float)(NumLake,
LakeNumComp,LakeCompNumStrm) ; Fraction of lake area impacted by the stream
LakeCompNumLake(Integer)
(NumLake,LakeNumComp) ; Number of lakes that impact lake compartment
LakeCompLakeIndex(Integer)(NumLake,
LakeNumComp,LakeCompNumLake) ; Index of lakes that impact lake compartment
LakeCompLakeFrac(Float)(NumLake,
LakeNumComp,LakeCompNumStrm) ; Fraction of lake area impacted by the lake

US EPA ARCHIVE DOCUMENT

LakeCompNumAqu(Integer)
(NumLake,LakeNumComp) ; Number of aquifer that impact lake compartment
LakeCompAquIndex(Integer)(NumLake,
LakeNumComp,LakeCompNumAqu) ; Index of aquifers that impact lake compartment
LakeCompAquFrac(Float)(NumLake,
LakeNumComp,LakeCompNumAqu) ; Fraction of lake area impacted by the aquifer
NumStrm(Integer) ; Number of streams
StrmId(String)(NumStrm) ; Environmental setting identification for streams
StrmName(String)(NumStrm) ; Name of streams
StrmLoc(GeoreferencePoint)(NumStrm) ; Location of streams
StrmNumSeg(Integer)(NumStrm) ; Number of stream segments
StrmSegLoc(GeoReferencePoint)
(NumStrm,StrmNumSeg) ; Location of stream segments
StrmSegNumStrm(Integer)
(NumStrm,StrmNumSeg) ; Number of streams that impact stream segment
StrmSegNumAqu(Integer)(NumStrm,
StrmNumSeg) ; Number of aquifers that impact stream segment
StrmSegAquIndex(Integer)(NumStrm,
StrmNumSeg,StrmSegNumAqu) ; Index of aquifer that impacts stream segment
StrmSegAquFrac(Float)(NumStrm,
StrmNumSeg,StrmSegNumStream) ; Fraction of stream segment impacted by aquifer
StrmNumWS(Integer)(NumStrm) ; Number of watersheds that impact stream
StrmWSIndex(Integer)
(NumStrm,StrmNumWS) ; Index of watershed that impacts stream
StrmWSFrac(Float)(NumStrm,,StrmNumWS) ; Fraction of stream segment impacted by the watershed

Type Air Layout
NumAir(Integer) ; Number of air angular segments
AirId(String)(NumAir) ; Environmental setting identification for air angular segment
AirName(String)(NumAir) ; Name of air angular segment
AirLoc(GeoReferencePoint)(NumAir) ; Center point of angular segment
AirWidth(Float)(NumAir) ; Width in degrees of angular segment
AirDelRadius(Float)(NumAir) ; Change in radius from inner radius to outer radius, in meters
; Assumed the same for each site as a part of assessment strategy
AirNumSrc(Integer)(NumAir) ; Number of sources impacting an air angular segment
AirSrcIndex(Integer)(NumAir, AirNumSrc) ; Index of sources impacting an air angular segment

Type Watershed Layout
NumWSReg(Integer) ; Number of watershed regions
WSId(String)(NumWSReg) ; Environmental setting identification for watershed regions
WSName(String) ; Name of watershed
WSLoc(GeoReferencePoint)(NumWSReg) ; Location of watershed
WSNumSub(Integer)(NumWSReg) ; Number of sub-basins in watershed
WSSubLoc(GeoReferencePoint)(NumWSReg,
WSNumSub) ; Location of sub-basins in watershed

WSSubArea(Float)(NumWSReg, WSNumSub) ; Area of sub-basin in square meters
WSSubNumAir(Integer)(NumWSReg, WSNumSub) ; Number of air angular segments that impact sub-basin
WSSubAirIndex(Integer)(NumWSReg, WSNumSub,WSSubNumAir) ; Index of air angular segment
WSSubAirFrac(Float)(NumWSReg, WSNumSub,WSSubNumAir) ; Fraction of sub-basin area impacted by air angular segment

WSSubNumSrc(Integer)(NumWSReg, WSNumSub) ; Number of sources that impact sub-basin
WSSubSrcIndex(Integer)(NumWSReg, WSNumSub,WSSubNumSrc) ; Index of source that impacts sub-basin
WSSubSrcFrac(Float)(NumWSReg, WSNumSub,WSSubNumSrc) ; Fraction of sub-basin area impacted by the source

Type Aquatic Foodchain Layout
NumAquaFC(Integer) ; Number of aquatic foodchains
AquaFCId(String)(NumAquaFC) ; Environmental setting identification for aquatic foodchain

AquaFCName(String)(NumAquaFC) ; Name of aquatic foodchain
AquaFCLoc(GeoreferencePoint(NumAqua) ; Location of aquatic foodchain
AquaFCNumStrm(Integer)(NumAquaFC) ; Number of streams that impact aquatic foodchain
AquaFCStrmIndex(Integer)(NumAquaFC, AquaFCNumStrm) ; Index of stream that impacts aquatic foodchain
AquaFCStrmFrac(Float)(NumAquaFC, AquaFCNumStrm) ; Fraction of aquatic foodchain impacted by stream
AquaFCNumLake(Integer)(NumAquaFC) ; Number of lakes that impact aquatic foodchain
AquaFCLakeIndex(Integer)(NumAquaFC, AquaFCNumLake) ; Index of lake that impacts aquatic foodchain
AquaFCLakeFrac(Float)(NumAquaFC, AquaFCNumLake) ; Fraction of aquatic foodchain impacted by lake

Type Terrestrial Foodchain Layout
NumTerrFC (Integer) ; Number of terrestrial foodchains
TerrFCId(String)(NumTerrFC) ; Environmental setting identification for terrestrial foodchain

TerrFCName(String)(NumTerrFC) ; Name of terrestrial foodchain
TerrFCLoc (GeoReferencePoint)(NumTerrFC) ; Location of terrestrial foodchain
TerrFCArea(Float)(NumTerrFC) ; Area of terrestrial foodchain in square meters
TerrFCNumAir(Integer)(NumTerrFC) ; Number of angular segments that impact terrestrial foodchain

TerrFCAirIndex(Integer) (NumTerrFC,TerrFCAir) ; Index of angular segment that impacts terrestrial foodchain

TerrFCAirFrac(Float) (NumTerrFC,TerrFCAir) ; Fraction of terrestrial foodchain impacted by air angular segment
TerrFCNumWS(Integer)(NumTerrFC) ; Number of watersheds that impact terrestrial foodchain

US EPA ARCHIVE DOCUMENT

TerrFCWSIndex(Integer)
(NumTerrFC,TerrFCWS) ; Index of watershed that impacts terrestrial foodchain
TerrFCWSFrac(Float)(NumTerrFC,TerrFCWS) ; Fraction of terrestrial foodchain impacted by watershed
TerrFCNumStrm(Integer)(NumTerrFC) ; Number of streams that impact terrestrial foodchain
TerrFCStrmIndex(Integer)
(NumTerrFC,TerrFCStrm) ; Index of stream that impacts terrestrial foodchain
TerrFCStrmFrac(Float)
(NumTerrFC,TerrFCStrm) ; Fraction of terrestrial foodchain impacted by stream
TerrFCNumLake(Integer)(NumTerrFC) ; Number of lakes that impact terrestrial foodchain
TerrFCLakeIndex(Integer)
(NumTerrFC, TerrFCLake) ; Index of lake that impacts terrestrial foodchain
TerrFCLakeFrac(Float)
(NumTerrFC,TerrFCLake) ; Fraction of terrestrial foodchain impacted by lake
TerrFCNumAquaFC(Integer)(NumTerrFC) ; Number of aquatic foodchains that impact terrestrial foodchain
TerrFCAquaFCIndex(Integer)
(NumTerrFC, TerrFCAquaFC) ; Index of aquatic foodchain that impacts terrestrial foodchain
TerrFCAquaFCFrac(Float)
(NumTerrFC,TerrFCAquaFC) ; Fraction of terrestrial foodchain impacted by aquatic foodchain
TerrFCNumAqu(Integer)(NumTerrFC) ; Number of aquifers wells that impact terrestrial foodchain

Type Farm Foodchain Layout
NumFarm (Integer) ; Number of farm or crop areas
FarmName(String)(NumFarm) ; Name of farm or crop area
FarmLoc (GeoReferencePoint)(NumFarm) ; Location of farm or crop area
FarmArea(Float)(NumFarm) ; Area of farm or crop area in square meters
FarmNumAir(Integer)(NumFarm) ; Number of angular segments that impact farm or crop area
FarmAirIndex(Integer)(NumFarm, FarmAir) ; Index of angular segment that impacts farm or crop area
FarmAirFrac(Float)(NumFarm,FarmAir) ; Fraction of farm or crop area impacted by air angular segment
FarmNumWS(Integer)(NumFarm) ; Number of watersheds that impact farm or crop area
FarmWSIndex(Integer)(NumFarm, FarmWS) ; Index of watershed that impacts farm or crop area
FarmWSFrac(Float)(NumFarm,FarmWS) ; Fraction of farm or crop area impacted by watershed

FarmNumStrm(Integer)(NumFarm) ; Number of streams that impact farm or crop area
FarmStrmIndex(Integer)(NumFarm, FarmStrm) ; Index of stream that impacts farm or crop area
FarmStrmFrac(Float)(NumFarm,FarmStrm) ; Fraction of farm or crop area impacted by stream
FarmNumLake(Integer)(NumFarm) ; Number of lakes that impact farm or crop area
FarmLakeIndex(Integer)(NumFarm, FarmLake) ; Index of lake that impacts farm or crop area
FarmLakeFrac(Float)(NumFarm,FarmLake) ; Fraction of farm or crop area impacted by lake
FarmNumWell(Integer)(NumFarm) ; Number of wells impacting farm
FarmAquIndex(Integer)(NumFarm) ; Index of aquifer that impacts farm or crop area
FarmWellIndex(Integer)(NumFarm,
FarmNumWell) ; Index of aquifer well that impacts farm or crop area

US EPA ARCHIVE DOCUMENT

FarmWellFrac(Float)(NumFarm,FarmNumWell) ; Fraction of farm or crop area impacted by aquifer well

Type Human Exposure Layout

NumHumRcp(Integer) ; Number of human receptors

HumRcpId(String)(NumHumRcp) ; Environmental setting identification for human receptor

HumRcpName(String)(NumHumRcp) ; Name of human receptor

HumRcpType(String)(NumHumRcp) ; Type of human receptor, e.g., “Resident Adult”,”Farmer Adult”,”Resident Child”,”Resident Worker”; a particular modeling route is turn off by setting the corresponding number of that media to 0. For example if inhalation was not wanted for a receptor the HumRcpNumAir for the receptor would be set to 0.

HumRcpNumFarm(Integer)(NumHumRcp) ; Number of farms that impact receptor

HumRcpFarmIndex(Integer)
(NumHumRcp,HumRcpNumFarm) ; Index of farm that impacts receptor

HumRcpFarmFrac(Float)
(NumHumRcp,HumRcpNumFarm) ; Fraction of human receptor area that farm impacts

HumRcpNumAquaFC(Integer)(NumHumRcp) ; Number of aquatic foodchains that impact receptor

HumRcpAquaFCIndex(Integer)
(NumHumRcp,HumRcpNumAquaFC) ; Index of aquatic foodchain that impacts receptor

HumRcpAquaFCFrac(Float)(NumHumRcp,
HumRcpNumAquaFC) ; Fraction of human receptor area impacted by aquatic foodchain impacts

HumRcpNumWell(Integer)(NumHumRcp) ; Number of well locations that impact receptor

HumRcpAquIndex(Integer)
(NumHumRcp,HumRcpNumWell) ; Index of aquifer that impacts receptor

HumRcpWell Index (Integer)(NumHumRcp,
HumRcpNumWell) ; Index of well that impacts receptor for the given aquifer

HumRcpWellFrac(Float)(NumHumRcp,
HumRcpNumWell) ; Fraction of human receptor area that aquifer well impacts

HumRcpNumStrm(Integer)(NumHumRcp) ; Number of stream segments that impact receptor

HumRcpStrmIndex(Integer)
(NumHumRcp,HumRcpNumStrm) ; Index of stream that impacts receptor

HumRcpStrmFrac(Float)
(NumHumRcp,HumRcpNumStrm) ; Fraction of human receptor area that stream impacts

HumRcpNumLake(Integer)(NumHumRcp) ; Number of lakes that impact receptor

HumRcpLakeIndex(Integer)(NumHumRcp,
HumRcpNumLake) ; Index of lake that impacts receptor

HumRcpLakeFrac(Float)
(NumHumRcp,HumRcpNumLake) ; Fraction of human receptor area that lake impacts

HumRcpNumWS(Integer)(NumHumRcp) ; Number of watersheds that impact receptor

HumRcpWSIndex(Integer)
(NumHumRcp,HumRcpNumWS) ; Index of watershed that impacts receptor

HumRcpWSFrac(Float)
(NumHumRcp,HumRcpNumWS) ; Fraction of human receptor area that watershed impacts

US EPA ARCHIVE DOCUMENT

HumRcpNumAir(Integer)(NumHumRcp) ; Number of air angular segments that impact receptor

HumRcpAirIndex(Integer)
(NumHumRcp,HumRcpNumAir) ; Index of air angular segment that impacts receptor

HumRcpAirFrac(Float)(NumHumRcp,
HumRcpNumAir) ; Fraction of human receptor area that air angular segment impacts

HumRcpNumSrc(Integer)(NumHumRcp) ; Number of sources impacting receptor

HumRcpSrcIndex(Integer)
(NumHumRcp,HumRcpNumSrc) ; Index of sources impacting receptor

HumRcpSrcFrac(Float)(NumHumRcp,
HumRcpNumSrc) ; Fraction of human receptor area that sources impact

Type Ecological Risk Layout

NumEcoRcp(Integer) ; Number of ecological receptors; a particular modeling route is turn off by setting the corresponding number of that media to 0. For example if terrestrial impacts were not wanted for a receptor, the EcoRcpNumTerrFC for the receptor would be set to 0.

EcoRcpId(String)(NumEcoRcp) ; Environmental setting identification for ecological receptor

EcoRcpName(String)(NumEcoRcp) ; Name of ecological receptor

EcoRcpNumTerr(Integer)(NumEcoRcp) ; Number of terrestrial foodchains that impact receptor

EcoRcpTerrIndex(Integer)
(NumEcoRcp,EcoRcpNumTerrFC) ; Index of terrestrial foodchain that impacts receptor

EcoRcpTerrFrac(Float)
(NumEcoRcp,EcoRcpNumTerrFC) ; Fraction of ecological receptor area that terrestrial foodchain impacts

EcoRcpNumAquaFC(Integer)(NumEcoRcp) ; Number of aquatic foodchains that impacts receptor

EcoRcpAquaFCIndex(Integer)
(NumEcoRcp,EcoRcpNumAquaFC) ; Index of aquatic foodchain that impacts receptor

EcoRcpAquaFCFrac(Float)
(NumEcoRcp,EcoRcpNumAquaFC) ; Fraction of ecological receptor area that aquatic foodchain impacts

US EPA ARCHIVE DOCUMENT

A.2 Procedure for Setting up Test Cases

A.2.1 Testing Module Sequencing

To set up a test case to test computational module sequencing, open the Windows® Explorer, go to the test bed directory, and follow these steps:

1. Make a copy of batch file run, naming it for the test case running (for example, MMSP_01 or MMSP_02).
2. Right-mouse-click on the new copied file, select edit.
3. Change the second line of the file commands to make the hd.ssf name match the test case (for example, hd111111.ssf for test case MMSP_01).
4. Save the file and exit.

A.2.2 Changing the Header File to Match the Test Case

Each test case must have a matching header file. To set up this file, open the Windows® Explorer, go to the test bed directory, and follow these steps:

1. Make a copy of the file hdexample.ssf, naming it for the test case you identified in Step 3 in Section A.2.1 (for example, hd111111.ssf for test case MMSP_01).
2. Double-click on the new file to open.
3. Change the following parameters:
   --"hdtest.ssf"--change to the name used in Step 1 of this section (for example, hd111111.ssf for test case MMSP_01).
   --"SSFDirectory"--change value immediately below it to the directory in which the test bed has been placed, identifying the subdirectory in which the SSF will be placed.
   --"GRFDirectory"--change value immediately below it to the directory in which the test bed has been placed, identifying the subdirectory in which the GRF will be placed.
   --"MMSP"--change value immediately below it to the directory in which the MMSP executable has been placed.
   --"Permanent"--change value immediately below it to the directory in which the test bed has been placed, identifying the subdirectory in which the permanent files will be placed.
   --"SiteID"--change value immediately below it to the same number used to identify the header file (for example, 111111 for hd111111.ssf and test case MMSP_01).
   --"Source"--change value immediately below it to whichever source type is being modeled (in essence, lf for landfill, lau for land application unit, si for surface impoundment, at for aerated tank, and wp for waste pile).
   --"StorageLevel"--change the value immediately below it to a zero (0) if minimal files will be saved (in essence, only the GRF) or one (1) if all files are to be saved to permanent storage.
   --"StopOnError"--change the value immediately below it to one (1) if the MMSP should stop when error conditions occur and zero (0) if the MMSP should continue when error conditions occur.
   --"StopOnWarning"--change the value immediately below it to a one (1) if the MMSP should stop when warning conditions occur or zero (0) if the MMSP should continue when error conditions occur.
4. Save the file and exit.

US EPA ARCHIVE DOCUMENT

A.2.3 Changing Site Layout Data to Match Test Cases

A Site Layout Data Group must also be created for each test case. To create such a file, open the Windows® Explorer, go to the test bed directory, and follow these steps:

1. Follow the paths to open the SSF directory.
2. Make a copy of file slexample.ssf, naming it according to what is in the test plan for that test case input (for example, slla111111.ssf for test case MMSP_01 in which the source being modeled is a land application unit).
3. Open the new file using the Wordpad.
4. To customize for your test case, search for the term "num." Modify those parameters found by increasing or decreasing, based on your site layout. Remember that if you change one parameter, you need to change others associated with it. For example, if you set the number of Water Body Networks to zero, you will need to set any other water body parameters to zero as well. Note also that you must have at least one farm for the human exposure module to receive input.
5. Save the file with its new name and exit.

US EPA ARCHIVE DOCUMENT