TOTAL PERFORMANCE ASSESSMENT (TPA) CODE VERSION 3.0 SOFTWARE REQUIREMENTS DESCRIPTION

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

Prepared by

Randall Manteufel

TOTAL PERFORMANCE ASSESSMENT (TPA) CODE VERSION 3.0 SOFTWARE REQUIREMENTS DESCRIPTION

1 INTRODUCTION

This software requirements description is the first step in updating the Total Performance Assessment (TPA) code from version 2.0 to 3.0. The TPA code Version 2.0 (Sagar and Janetzke, 1993) was used in the Nuclear Regulatory Commission/Center for Nuclear Waste Regulatory Analyses (NRC/CNWRA) Iterative Performance Assessment (IPA) Phase 2 exercise. The TPA code is an executive module and a set of consequence modules that simulate the performance of a geologic repository of nuclear high-level waste (HLW) at Yucca Mountain (YM), Nevada. The executive module controls the flow of data and execution between the process/component-specific consequence modules that simulate major safety components of the repository system. The TPA code integrates geologic site characterization data, proposed repository and waste package (WP) engineered designs, and biosphere data. The consequence modules are designed with input from materials engineers, hydrogeologists, seismologists, volcanologists, rock mechanicians, and health physicists. Recent developments in the proposed YM repository necessitate a new version of TPA code. These developments include (i) new repository, WP, and emplacement designs, (ii) changing regulatory standards (from release-based to dose-based), and (iii) potentially longer time periods of concern (to hundreds of thousands of years). Numerous improvements will be incorporated that reflect knowledge and data gained in recent years of site characterization and laboratory studies, as well as other total system performance assessments (TSPA). The TPA code will have the capability to assess the compliance of the proposed YM repository with regulatory requirements using a probabilistic approach to account for uncertainties.

2 SOFTWARE FUNCTION

The TPA code is a combination of an executive module and a set of consequence modules that stochastically assess the overall performance of the proposed YM HLW repository with applicable regulatory standards. The executive driver controls the probabilistic sampling of input parameters, the calculational flow process between modules, and the generation of output files. Output files can be used for parameter importance analyses, generation of time-dependent risk curves, and generation of complementary cumulative distribution functions for cumulative release of radionuclides. Utility modules ensure a consistent description of the proposed repository system and flow of data between consequence modules. Examples include the spatial and temporal discretizations (i.e., subarea (SA) discretization of the proposed repository and time stepping scheme). In the NRC/CNWRA IPA Phase 2 exercise, there were 7 SAs and 50 time steps of 200 yr each. The number of SAs in the TPA code will be based on the latest proposed repository design and reflect near-field thermal-hydrologic-mechanical-chemical (THMC) environments in the proposed repository as well as hydrostratigraphy. The time stepping will be variable over the simulation as well as the total time period of interest (TPI).

3 TECHNICAL AND COMPUTATIONAL APPROACHES

The overall conceptual approach of a TSPA is outlined in figure 1. Data flow from the system characterization to final regulatory compliance determination (shown as either a cumulative release or dose). The bulk of the modeling effort is in the consequence modules that include both anticipated

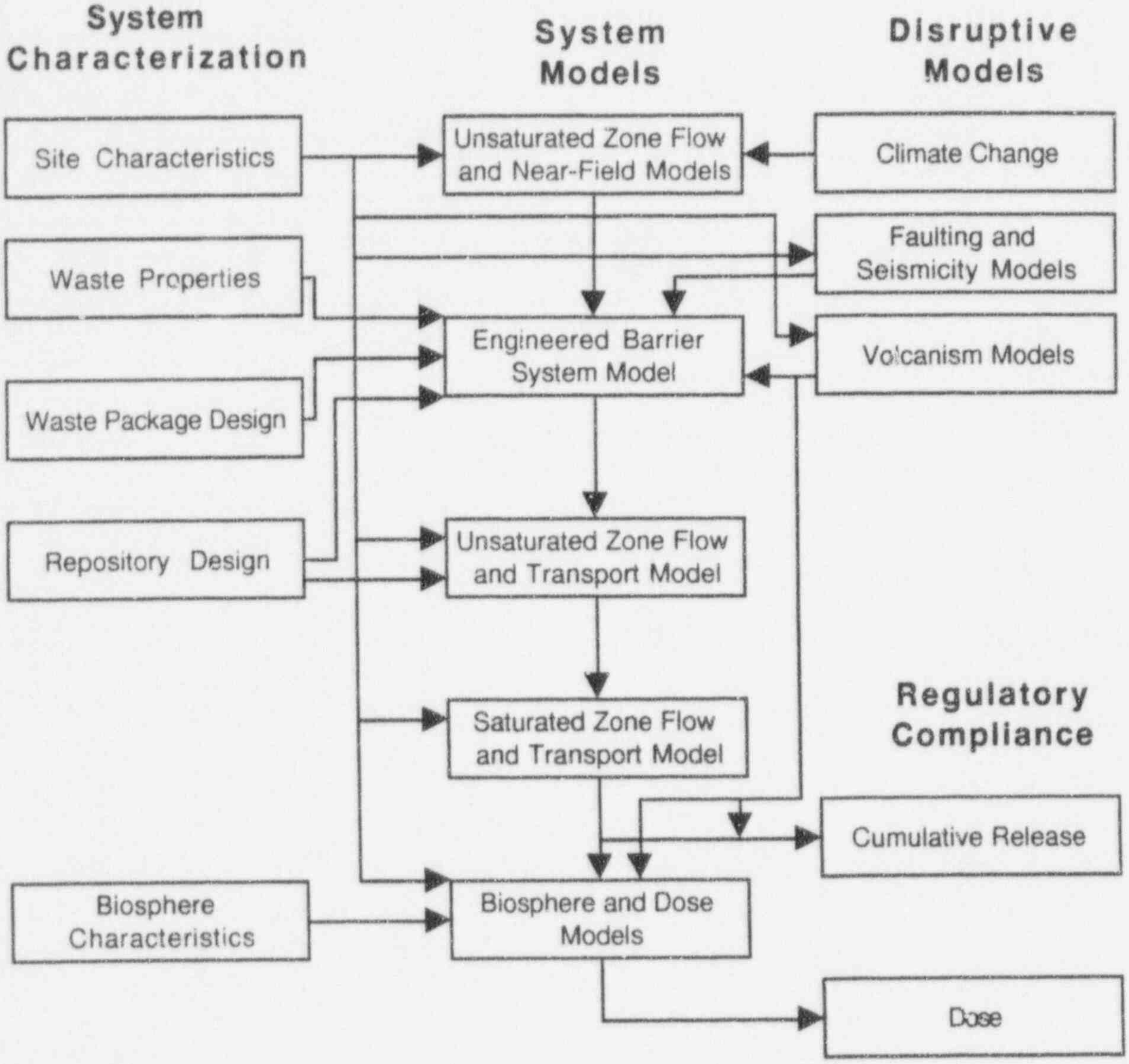

Figure 1. Overall TSPA flow diagram

processes (also called base-case processes) and disruptive processes. The base-case system has seven major subsystem models:

- groundwater flow from the ground surface to the proposed repository,
- near-field THMC environment of the engineered barrier system (EBS),
- corrosion and other anticipated failure mechanisms of the EBS containment,
- release of radionuclides from the EBS into geologic setting,
- groundwater flow and radionuclide transport in the unsaturated zone below the proposed repository and into the saturated zone,
- groundwater flow and radionuclide transport in the saturated zone below the proposed repository to a compliance point (CP) or boundary, and
- transport of radionuclides in the biosphere through the groundwater pathway that leads to dose to humans.

The disruptive system has faulting, seismicity, volcanism and climate change that cause earlier failures of EBS containment. In the case of volcanism, radionuclides may be released directly into the accessible environment and at the CP through the ground surface pathway.

A number of utility modules will be used to provide general data and generic capabilities that more than one module may need. For example, the initial radionuclide inventory will be calculated in a module so this information can be provided to other modules.

The TPA code will control the spatial discretization of the proposed repository (i.e., number of SAs), the distance from the proposed repository to the CP (e.g., 5, 20, 25, or 30 km), the temporal discretization scheme (e.g., output every 200 yr), and the TPI (e.g., 100,000 yr).

Major analysis improvements in the TPA code include the ability to

- increase or decrease the TPI
- evaluate finer time discretizations using nonuniform time steps
- evaluate finer repository spatial discretizations
- evaluate different areal mass loading
- calculate time-dependent dose rate at a CP
- calculate peak dose rate at a CP in the TPI
- evaluate dilution in saturated zone
- evaluate in-drift emplacement design
- add or remove sampled parameters

The TPA code Version 3.0 will also include:

- updated consequence models
- improved parameter importance analysis capabilities
- streamlined scope for consequence modules
- streamlined methodology for data transfer between consequence modules
- more flexible design to accommodate changes in consequence modules

4 USER INTERFACE AND DATA FLOW

The TPA code will be executed in file batch mode using one main input file: "tpa.inp." The TPA code reads data from this one input file only. The TPA code writes output data into a set of files for plotting or importance analyses. The file interfaces are described here.

4.1 INPUT TO TOTAL PERFORMANCE ASSESSMENT CODE

All of the input data for the TPA code is contained in the "tpa.inp" file. No other files/input will have an effect on the TPA executive (EXEC) calculations. Auxiliary files for data may be needed for consequence modules, however, these files should be "static" and the "tpa.inp" file should be used for parameter descriptions that change from run to run. The "tpa.inp" file contains all parameters necessary to describe the scenario and the number of realizations requested. The "tpa.inp" file starts with two comment lines that the analysts should use to describe the type of run being performed. These lines will be read as Character*80 and echoed at the top of all output files.

4.2 FLOW OF DATA BETWEEN MODULES

EXEC controls data flow by passing data in the subroutine call statement to each module. EXEC does not use common blocks or disk files for data transfer between itself and consequence modules. Within a consequence module, common blocks or files can be used. EXEC does not permit that data be passed directly between consequence modules. Each module is to be called only by EXEC and not by other modules. For efficiency and implementation purposes, the modules can consist of more than one subroutine, may call TPA code utility subroutines (e.g., INVENT), or may call stand alone programs (e.g., NEFTRAN). But modules are to pass information only to EXEC to control the simulation process.

The overall sequence of execution and flow of data is shown in figure 2. Here, EXEC starts the simulation by reading the "tpa.inp" file through the READER routine. The READER module need only be called once during a run. Having determined the parameters that describe the system, the EXEC continues by calling component specific consequence modules. Some modules will only be called once during one realization, while others will be called many times. Modules being called once include SAMPLER, SZFT, DCAGW and DCAGS. The modules UZFLOW, NFENV, EBSFAIL, EBSREL, and UZFT will be called once for each SA for each realization. If disruptive scenarios are being analyzed, the FAULTING, SEISMO, VOLCANO, ASHPLUME, and DCAGS modules will be called directly by EXEC once during a realization. These disruptive modules will be used to either cause earlier failure of the EBS or provide a more direct pathway for radionuclides into the biosphere (e.g., VOLCANO through ASHPLUME). If desired in the future, disruptive scenarios can be designed or modified to affect groundwater flow and transport of radionuclides.

The computational scheme for the TPA code is shown in figure 3. Each module is called by the EXEC module with some inputs and then returns some outputs to the EXEC through the call statement. Consequence modules do not call each other directly. There is a clear expectation of inputs and outputs between the EXEC and each module. In most cases, the output of one module is the input to the next module. The EXEC has two main loops for the number of realizations and the number of SAs.

The utility modules and consequence modules are discussed next.

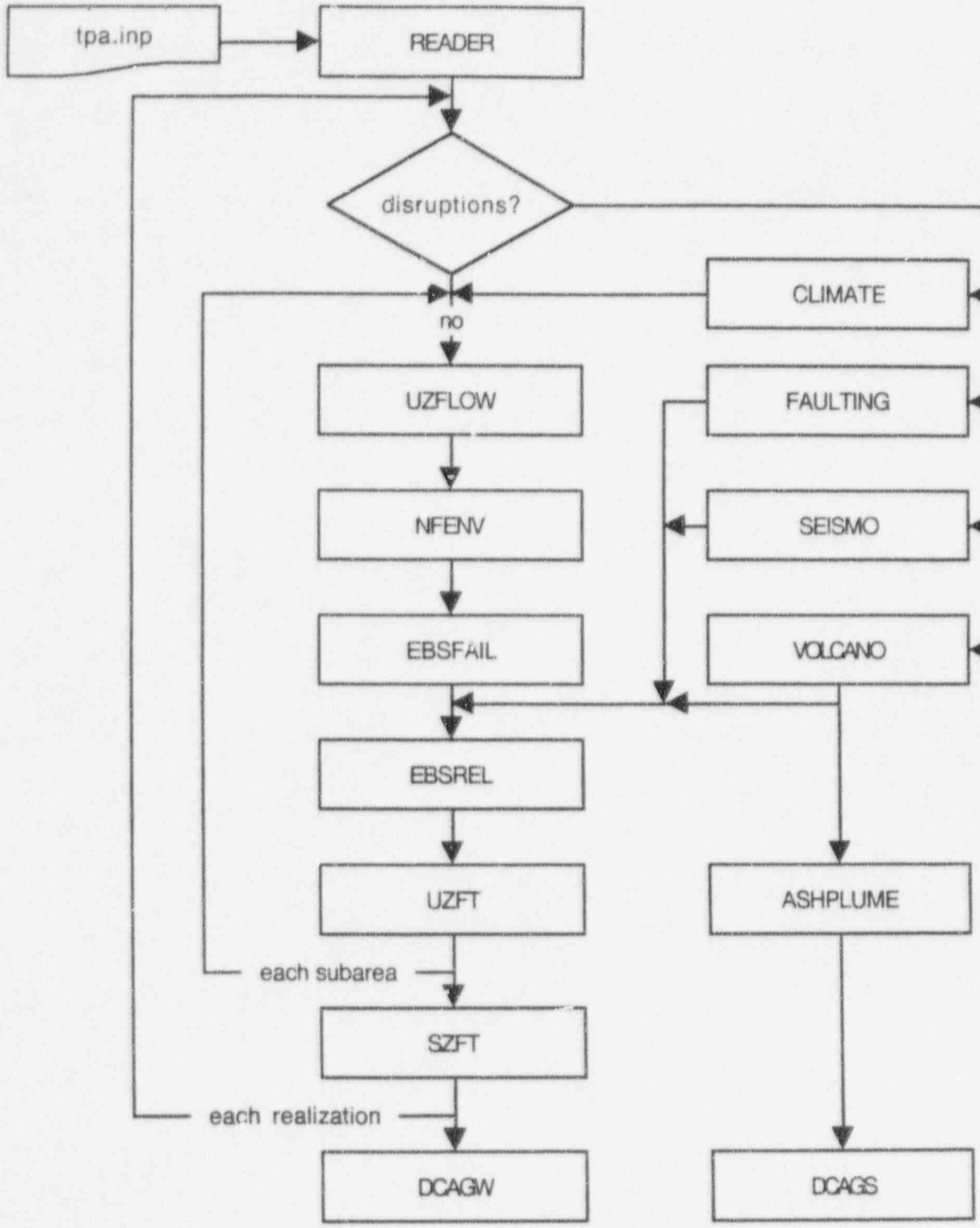

Figure 2. TPA flow diagram

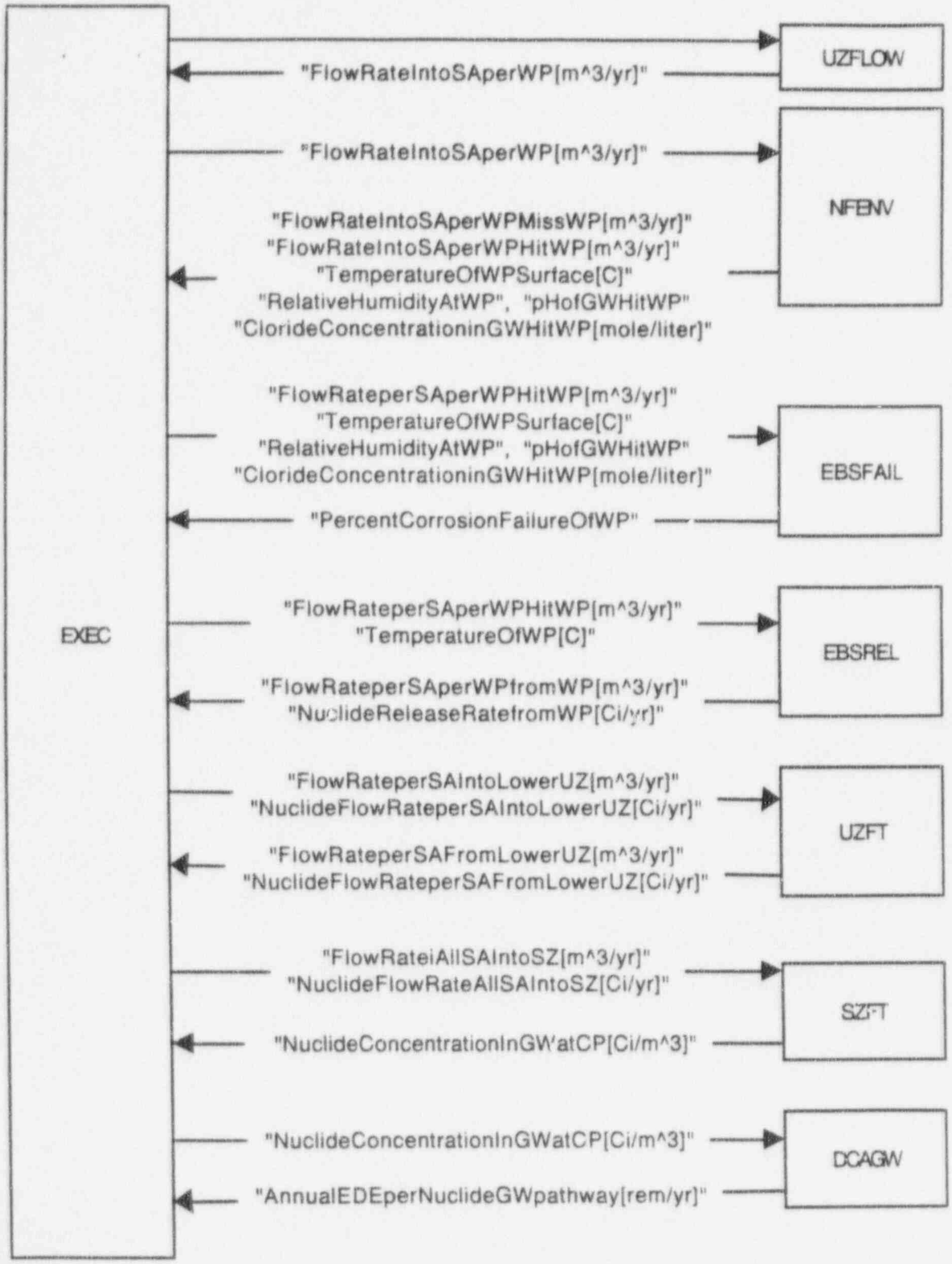

Figure 3. Main input/output associated with base-case flow and transport

4.3 UTILITY MODULES

4.3.1 Reader

READER is a utility module that preprocesses the data from the "tpa.inp" file. This is the only subroutine that reads the "tpa.inp" file. This module is similar to that already existing in the TPA code Version 2.0. The "tpa.inp" file will contain data specific for the TPA code execution as well as all probability distribution function (PDF) definitions for parameters that will be provided to the consequence modules.

4.3.2 Sampled Parameter

This module dynamically stores and retrieves information associated with parameter probability density functions (PDFs). PDFs are read from the "tpa.inp" file during run time. The number of distributions includes: CONSTANT, UNIFORM, LOGUNIFORM, NORMAL, LOGNORMAL, BETA, LOGBETA, TRIANGULAR, LOGTRIANGULAR, and EXPONENTIAL. In addition joint PDFs relating two parameters will be supported using a correlation matrix approach. All PDFs will be sampled for each of the realizations required in the simulation.

4.3.3 Invent

INVENT is a utility module that allows centralized computation and storage of radionuclide inventory data. This module is a set of subroutines based on the subroutines described in Lozano et al. (1994). The subroutines provide the inventory (in Ci/MTU) of 43 radionuclides for times to one million years.

4.3.4 Subarea

SUBAREA is a utility module for the storage and retrieval of repository SA information. The database is created once in the READER module and then the information will be available to all other modules. The consequence modules can acquire information about the SA discretization, but not change the information.

4.4 CONSEQUENCE MODULES

The main consequence modules in the TPA code are UZFLOW, NFENV, EBSFAIL, FAULTING, SEISMO, VOLCANO, EBSREL, UZFT, SZFT, ASHPLUME, DCAGW, and DCAGS. These modules will interface with EXEC using a subroutine call statement. The parameters and arrays being passed in the call statement are negotiated between the EXEC and each of the modules. Consistent with the software design principle of procedural abstraction, the EXEC does not need to know how the calculations are performed in the consequence modules. There are at least four ways of doing the calculations: abstraction of results (i.e., table look-up), abstraction of models, incorporation of the main calculational routines from an existing code, or spawning an independent process that executes a stand alone code. Previously, the EXEC explicitly called routines and spawned processes. In the new EXEC, main consequence modules are called directly and the I/O between EXEC and modules need not be changed in the future if the implementation in the module changes.

. Because the TPA code simulates the time-dependent response of the proposed repository, it decides the overall time discretization to be used for all consequence modules. The time discretization is intended to synchronize input and output between modules and should not be confused with timesteps used in solving transient problems in various modules. The time discretization can and will often be nonuniform, especially for simulations out to hundreds of thousands of years. An example of a time discretization is {0, 10, 25, 50, 75, 100, 125, 150, 200, 250, 300, ..., 9,500, 10,000} yr. All of the time-dependent inputs and outputs for the consequence modules must be provided at these time steps.

The UZFLOW module will provide estimates of percolating flow rates into the near-field of the proposed repository. Separate flow rates will be estimated for each of the proposed repository SAs. For example, if six SAs are used for the proposed repository, then UZFLOW will be called six times to provide estimates of flow into the near-field. The flow rates are time-dependent and need to be predicted for the times provided by the TPA code. This module may account for long-term trends that affect percolation (e.g., climate changes) or short-term changes (e.g., abnormal wet period). Output data from the UZFLOW module will be input to the NFENV module.

The NFENV module will provide estimates of near-field conditions for each proposed repository SA. The NFENV module should account for the location of the SA (interior or edge regions). The output of the NFENV module is the near-field rock temperature, WP surface temperature, spent fuel temperature, relative humidity at the WP, flow rates into the EBS, and geochemical condition of groundwater flowing into the EBS. All of this output will be time-dependent. These data will be provided to EBSFAIL module.

The EBSFAIL module uses the output of the NFENV module to predict failure of the EBS to contain waste. Failure can be the result of corrosion, as well as other anticipated causes. Examples of anticipated causes of failure include initial defects, thermal-induced stresses in the WP, and anticipated seismic activity. A separate SEISMO module will also evaluate the consequences of seismic activity. At this time, EBSFAIL should account for the numerous small magnitude events while SEISMO should evaluate low-probability, high-consequence events. Possibly, EBSFAIL should evaluate activity with recurrence intervals of up to 500 - 1,000 yr, and SEISMO should evaluate stronger events that have longer recurrence intervals. EBSFAIL and SEISMO analysts need to negotiate this detail. Low-probability, high-consequence disruptive causes of failure are not considered in EBSFAIL. The primary output of EBSFAIL will be a time-dependent fraction of EBS failure to contain the waste. The fraction may start at a nonzero value due to initial defects and may not reach 100 percent within maximum simulation time.

The FAULTING, SEISMO, and VOLCANO modules each predict failure of the EBS due to disruptive processes and events or additional mechanical loads for the WP that accelerate failure of the EBS. These modules are called only for disruptive scenarios and not for the base-case scenario. Each module generates a time-dependent failure curve for WPs in each realization. These failures are combined by the TPA code with the EBSFAIL failures to have an overall percent failure.

The EBSREL module predicts the transient release rate [Ci/yr] of each radionuclide per WP in the SA. The radionuclides are released from the EBS and into the lower unsaturated zone region that extends from below the proposed repository to above the water table.

The UZFT module predicts release [Ci/yr] of each radionuclide from the unsaturated zone into the saturated zone. The module simulates gravity-driven percolating flow and radionuclide transport in

the fractured stratified hydrogeology. The module accounts for the retardation of nuclides. The sum of the releases from all SAs is then provided to the SZFT module.

The SZFT module predicts the transient groundwater source [Ci/m^3] of each radionuclide at the CP which may be a well located 5, 20, 25 or 30 km away from the proposed repository. The module accounts for the retardation of nuclides, plume dilution, and dilution due to the pumping rate at the well.

The ASHPLUME module provides an extra pathway for radionuclides to be transported into the biosphere. This pathway is due to extrusive volcanic events that entrain waste in the magma and spread the waste in the volcanic ash plume. After VOLCANO is called, ASHPLUME will be called to evaluate this pathway for waste.

The DCAGW module simulates the biosphere and computes dose rates [rem/yr] from the groundwater pathway. DCAGW uses the output of SZFT. The DCAGW module is based on the GENII code which has been applied to YM biosphere conditions.

The DCAGS module simulates the biosphere and computes annual doses [rem/yr] from ground surface pathways. DCAGS uses output from ASHPLUME. The DCAGS module is based on the GENII code which has been applied to YM biosphere conditions.

4.5 OUTPUT FROM TOTAL PERFORMANCE ASSESSMENT CODE

The TPA code will generate results that can be used in importance analyses and in assessment of proposed repository compliance with either dose, risk, or release-based standards. The output files generated are listed here:

sp.dat = input values sampled for each R from PDFs described in "tpa.inp"

mv.dat = constant (e.g., not time- or nuclide-dependent) module variables for each module for each R/SA that will be used for importance analyses

uzflow.dat = output values for each R/SA from the UZFLOW module

nfenv.dat = output values for each R/SA from the NFENV module

ebsfail.dat = output values for each R/SA from the EBSFAIL module

ebsrel.dat = output values for each R/SA from the EBSREL module

uzft.dat = output values for each R/SA from the UZFT module

szft.dat = output values for each R from the SZFT module

dcagw.dat = output values for each R from the DCAGW module

dcags.dat = output values for each R from the DCAGS module

doseavg.dat = annual effective dose equivalent at CP

ccdf.dat = data used to generated complementary cumulative distribution function for Environmental Protection Agency normalized release to accessible environment located at 5 km from the proposed repository over a 10,000 yr time period

The first two lines of any output file will echo the first two lines of the "tpa.inp" file. The third line of any output file will provide the version number of the TPA code being used and the time and date of the run. Data will follow on subsequent lines depending on the specific file.

5 PROGRAMMING LANGUAGES

The TPA code is written in FORTRAN 77 as implemented in the SUN SPARCompiler, Version 2.0. The length of variable names will not be restricted to seven or less characters, but will be as long as needed to readily identify the variable. In addition, some compiler specific calls for date and time will be used. Although not recommended, modules can be written in languages other than FORTRAN 77 or can be standalone computer programs. In these cases, the responsible programmers must provide a FORTRAN 77 interface subprogram consistent with the TPA-module interface described in the previous section.

6 HARDWARE PLATFORMS

The TPA code will be developed for execution on SUN machines using the UNIX operating systems. The code will be designed such that it will also run on other operating systems to the extent practical, such as the CRAY computer.

7 GRAPHICS OUTPUT

No special graphics devices will be supported. Output will be in the form of ASCII files written in a format that can be read by spreadsheet programs, analysis software, and plotting packages.

8 PRE AND POST-PROCESSOR

No pre or post-processor is required or supported by the TPA code. The output files generated by the TPA code will be designed so that they can be read easily by spreadsheet programs, analysis software, and plotting packages.

9 REFERENCES

Lozano, A.S., H. Karimi, J.P. Cornelius, R.D. Manetufel, and R.W. Janetzke. 1994. *INVENT: A module for Calculation of Radionuclide Inventories, Software description, and User Guide,* CNWRA 94-016. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

Sagar, B., and R.W. Janetzke. 1993. *Total-System Performance Assessment (TPA) Computer Code: Description of Executive Module, Version 2.0.* CNWRA 93-017. San Antonio, TX: Center For Nuclear Waste Regulatory Analyses.

Rationale for Changes from TPA Version 2 to TPA Version 3 Computer Code

Background:

The NRC TPA computer code (Sagar and Janetzke, 1993), designated version 2, was developed in 1993 for the NRC IPA Phase 2 Assessment (Wescott et al., 1994) of the proposed YM repository site. The TPA version 2 code was designed to calculate the performance measures specified in NRC 10 CFR 60 (i.e., container lifetime, release rate, groundwater travel time) and the EPA standard (i.e., cumulative release for 10,000 yrs). Additionally, the code calculates integrated population dose over 10,000 yrs. Since the initial development, there have been no major upgrades to the TPA version 2 code (although attempts were made to improve the code computational efficiency). NUREG-1464 (Wescott et al., 1994) documents recommendations for improvement of the TPA version 2 code including: (i) software QA requirements need more prominence in module development, (ii) future IPA developments will require more model abstractions and efficient computing techniques, and (iii) the TPA code must be easily upgraded. With regards to the latter recommendation, NUREG-1464 states that the TPA system code "be considered a dynamic entity, to be upgraded in future IPA iterations. Possible upgrades include: addition of new modules, changed scope of current modules, centralized use of databases, uniform interfaces between modules, and uniform coding practices among modules." Most of these recommendations have been followed in developing TPA version 3, particularly those related to software QA.

Major revisions of the TPA code were required to address: (i) recommendations of the National Academy of Sciences (which will require the EPA to establish new performance criteria), (ii) changes in the DOE repository design, (iii) expanded knowledge base, improved models, and additional data compiled by NRC/CNWRA, and (iv) specific needs for the KTI sensitivity analyses. In addition, recent applications of the TPA version 2 identified certain software QA vulnerabilities and some problems with regard to the lack of flexibility of the code to accommodate new requirements and to allow simple modifications.

The recommendations of the NAS have required the following changes in Total-System Performance Assessment (TSPA) methodology for the proposed repository: (i) addition of a new performance measure (individual dose versus cumulative release), (ii) addition of variable compliance period (time to peak dose, not necessarily 10,000 yrs), (iii) elimination of human intrusion by drilling (not considered part of the main compliance determination), and (iv) incorporation of a more appropriate representation of environmental pathways and dose.

DOE repository design changes that necessitated TPA code modifications include waste package design (e.g., in-drift emplacement of large packages versus vertical emplacement of small packages) and areal heat loading (which determines the area requirements and layout of the repository blocks). The NRC technical assistance studies and former research projects have produced technical bases for improving various modules of the TPA code. Some of the major technical contributions are:

- The former Container Lifetime and Source Term (CLST) KTI produced an improved source term module.

- Igneous Activity (IA) KTI has produced improved technical basis for the probability model and scenario characteristics for simulating volcanic eruptions.

- Structural Deformation and Seismicity (SDS) KTI has produced an improved hydrostratigraphic model and technical specifications for a new consequence module to simulate a faulting scenario.

- Thermal Effects on Flow (TEF) KTI has provided improved predictions of temperature and relative humidity (for use in the source term module).

- Unsaturated and Saturated Flow under Isothermal Conditions (USFIC) KTI has produced improved information on climate and the distribution of infiltration over the repository area.

In the NRC/CNWRA audit review (Baca and Brient, 1996) of the DOE TSPA-95 report, the TPA Version 2 code was run to independently check selected DOE CCDFs for cumulative release. In this application, an attempt was made to modify the TPA Version 2 code to accommodate compliance periods longer than 10,000 yrs. It was found that this simple extension of time could not be accommodated without significantly modifying the source term module as well as other modules. Additional code limitations and problems were also identified. For example, it was found that certain input data to one module was not used because of undocumented "hard coding" in the module, the LHS module could not accommodate the beta (or log-beta) distribution used by DOE, and an error in units conversion was found in another module. Other limitations of the TPA version 2 code were associated with duplication of data files (creating potential consistency problems), numerous large data files created by the code (requiring computers with large disk space), and very long run times for individual scenarios.

Transition from TPA Version 2 to TPA Version 3:

The TPA Version 2 executive and modules, which were developed in 1993, are documented in Sagar and Jantezke (1993), Wescott et al. (1994) and in various consequence module users guides. The TPA executive is structured to control the sequence of execution of individual modules. Those modules consist of LHS (random parameter sampler), CANT2 (waste package temperature simulator), FLOWMOD (unsaturated zone flow), C14 (carbon-14 gas phase transport), SOTEC (source term code), NEFTRAN (radionuclide transport), CLIMATO (place holder for climate dependent infiltration rate), SEISMO (seismic impact on waste packages), DRILLO1 and DRILLO2 (human intrusion consequences), VOLCANO (volcanism consequences), AIRCOM (interface for airborne releases), and DITTY (population dose module).

The TPA version 3 Software Requirements Description (SRD) (Manteufel, 1997) provides detailed descriptions of the major code changes necessary to address the new performance criteria, design characteristics, and incorporate improved models. To ensure that the planned KTI sensitivity studies could be completed in a timely manner, it was considered vital that the new version of the TPA code be very transparent (i.e., easy to understand and follow the coding logic), have relatively short run times, and permit easy to access intermediate and final calculations. The following descriptions summarize the major revisions to the individual TPA modules and the technical reasons for the changes.

TPA Executive:

The TPA Version 2 executive or driver module was rewritten to: (i) provide greater transparency of the flow of data and calculational results between modules, and (ii) permit the user to have direct access to inputs and outputs for the conduct of sensitivity analyses. The TPA version 3 executive is about

500 lines of Fortran code versus the 20,000 lines of code in TPA version 2. The new version retains the TPA Version 2 algorithm for computing the CCDF, but uses a different approach for invoking the sequence of consequence module runs (i.e., subroutine calls rather than spawning processes).

LHS Sampler:

The new Sampler module retains the same sampling approach of the original LHS module but with the added capability for additional distributions (beta and log-beta) which are commonly used by DOE. In addition, the new module removes the ordering restriction on input parameters and enhances the transparency of parameter transfers from the sampler to the consequence modules.

CANT2 Module:

The CANT2 module will be upgraded by incorporating a 3D conduction only model. This upgraded version is being developed under the TEF KTI. In addition, time histories of thermal-chemical processes will be generated with MULTIFLO and added at a later time.

FLOWMOD Module:

Functionality of the FLOWMOD module will be retained in the new module called UZFLOW, and modified to reflect more recent information regarding infiltration distributions provided by USFIC KTI.

C14 Module:

This module has been eliminated because gas phase carbon-14 is not significant from an individual dose standpoint. TSPAs conducted by NRC and DOE have shown that the numerical limit for C-14 cumulative release (in the EPA 1985 standard) is overly stringent and inappropriately constrain the performance of the repository.

SOTEC Module:

This module has been replaced because of the availability of the new EBSPAC modules (EBSFAIL and EBSREL) which contains the most recent waste package design and improved models for container life and release calculations. The NRC staff have assumed responsibility for further development of the EBSREL module.

NEFTRAN Module:

This module has been retained with minor modification and is utilized in the new UZFT and SZFT modules. In consultation with the NRC staff, the module input/output and hydrostratigraphic data are currently being modified for incorporation in the new version of TPA.

CLIMATO Module:

This place holder module (renamed CLIMATE) will be used to incorporate a description of long-term changes in infiltration rates. The data and calculational approach will be provided by the USFIC KTI.

SEISMO Module:

This module will be superseded by an upgraded module currently being developed under the TSPA KTI (transferred from the former RDTME KTI). The development of the new module is required to accommodate the new waste package design and incorporate new models of seismicity and drift stability.

DRILLO1 and DRILLO2 Modules:

These two modules were eliminated because human intrusion by drilling will not be considered in compliance determinations, as recommended by the NAS. In addition, previous consequence analyses (see Annual Report) have indicated that this scenario has little impact on dose and cumulative release.

VOLCANO Module:

This module has been modified to incorporate the probability map and scenario characteristics provided by the IA KTI. In addition, the module was linked to a new ASHPLUME module that calculates ash/radionuclide dispersal from the simulated volcanic eruption.

AIRCOM Modules:

This module, which interfaces with DITTY, will be eliminated because it no longer required.

DITTY Modules:

This module will be eliminated because the population dose is no longer a relevant performance indicator (per the NAS recommendations for individual dose). DITTY will be replaced by look-up table modules DCAGW (dose factors for groundwater pathway) and DCAGS (dose conversion factors for ground surface pathways). Dose factors for DCAGW and DCAGS will be computed using the GENII dose code.

New TPA Modules and Utilities:

Three new modules have been added to permit consideration of relevant features, events, and processes. These new modules include: (i) FAULTING (calculates the impact of fault displacements on waste packages), (ii) ASHPLUME (calculates the ash and radionuclide dispersal resulting from a volcanic eruption through the repository), and (iii) NFENV (provides parameter specifications for the near-field environment). Various utility routines were developed to centralize data sets and calculational algorithms. For example, the INVENT utility module was developed to provide a centralized data set for the radionuclide inventory in the waste package as a function of time and spent fuel characteristics.

Summary:

The basic rationale for revising the TPA code was: (i) the need for a code suitable for use by a wide variety of users, (ii) the requirement to accommodate expected changes in the regulations (e.g., performance measures, compliance time period, etc), (iii) repository design changes (waste package design and emplacement geometry), and (iv) the need to incorporate the new and improved modules. Because the TPA code will now be used by many users, it was deemed vital that new version of the code

be highly transparent (with regards to its functionality), significantly easier to modify and use, and computationally more efficient. In developing the TPA version 3, the existing consequence modules that were relevant to the new requirements were placed under a new and more compact executive program. The consequence and utility modules that have been developed/modified to-date have been extensively tested for software QA purposes. Many of the improvements to the TPA code are consistent with the recommendations outlined in NRC IPA Phase 2 report (Wescott, et al., 1994). The design of the new TPA version 3 code is documented in a series of draft SRDs (Manteufel, 1997) which were provided to and discussed with the NRC staff, in meetings such as the one on November 11, 1996 (prior to the Annual Center Review meeting).

References:

Baca, R.G., and R.D. Brient (Editors). 1996. *Total System Performance Assessment 1995 Audit Review.* Letter report submitted to NRC. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

Manteufel, R.D. 1997. *Total Performance Assessment (TPA) Code Version 3.0 Software Requirements Description.* Letter report submitted to NRC. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

Sagar, B., and R. Janetzke. 1993. *Total-System Performance Assessment (TPA) Computer Code: Description of Executive Module, Version 2.0, Iterative Performance Assessment, Phase 2.* San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

Wescott, R.G., M.P. Lee, N.A. Eisenberg, T.J. McCartin, and R.G. Baca (Editors). 1994. *NRC Iterative Performance Assessment Phase 2.* NUREG-1464. Washington, DC: U.S. Nuclear Regulatory Commission.