

**SOFTWARE REQUIREMENTS SPECIFICATION
FOR THE PARCS-SPECIFIC DATA MAP ROUTINE
IN THE COUPLED RELAP5/PARCS CODE**

Douglas A. Barber, Thomas J. Downar

School of Nuclear Engineering
Purdue University
W. Lafayette, IN 47907-1290

submitted to:

Division of Systems Technology
Office of Regulatory Research
Nuclear Regulatory Commission
Washington, D.C. 20555-001

Contract NRC-04-97-039, Task 2
Performed Under Subcontract to SCIENTECH, Inc.

May 1998

I. Introduction

In the framework of the coupled RELAP5/PARCS code, PARCS is designed to be a self-contained process, which communicates with the General Interface process through the use of the constructs supplied with the Parallel Virtual Machine (PVM) package. In order to accommodate the requirements of the General Interface and minimize the coding modifications to PARCS, a separate data map routine is needed to manage the transfer of data between these two processes.

The primary responsibility of the PARCS-specific data map routine (PDMR) is to function as a secondary interface between the General Interface and PARCS. In order to couple PARCS to the General Interface, and thus RELAP5, the PDMR controls several tasks, the first of which relates to the processing of the permutation matrices previously described in the Software Requirements Specification (SRS) for the General Interface⁽¹⁾. Second, the PDMR manages the transfer of initial and time-dependent control information needed for calculational coherency between RELAP5 and PARCS. Finally, the PDMR controls both the processing and the transfer of the vectors of space-dependent property data both to and from the General Interface. The requirements needed to perform these tasks are outlined in this document, and are consistent with the requirements specified in the SRS for the RELAP5-Specific Data Map Routine (RDMR)⁽²⁾.

II. Software Requirements

The PDMR is a FORTRAN-90 subroutine called from PARCS, which handles all communication with the General Interface, and consequently translates the data between PARCS and the General Interface, both of which utilize SI units. To maintain consistency with the design requirements of the General Interface and RELAP5, the PDMR contains three functional units relating to initialization, mapping of data from the General Interface to PARCS, and mapping of data from PARCS to the General Interface. In addition, an error checking unit is included, and sufficient fault logic is provided for handling failures in the coupled code. In such instances, it is necessary to communicate information between each process in order to safely terminate the execution.

II.A. Functionality

The primary task of the PDMR is to control the communication of data between the General Interface and PARCS, and in doing so, allow the design of the General Interface to remain immutable. In addition, since it is necessary to reduce the required modifications to the PARCS source, the integration of the PDMR within PARCS is intended to be modular in nature. To accommodate this, the PDMR is designed to meet several requirements, as outlined in the following subsections.

II.A.1. Variable Requirements

The variable requirements of the PDMR consist primarily of the data type-dependent control buffers used to transmit initial and time-dependent control information, the permutation

matrices, the permuted vector of space-dependent thermal-hydraulic and heat structure data, and the unpermuted vector of space-dependent neutronic data. The nature of the initial and time-dependent control information is described in the following sections.

The variables representing the two permutation matrices, used to map the space-dependent property data between the RELAP5 and PARCS, utilize the Coordinate Storage Format⁽³⁾. Thus, the variables required relate to the vector of non-zero matrix values, and the vectors of row and column indices corresponding to these values. In order to describe the permutation matrices and the vectors of space-dependent property data, structure and dimension variables are also needed.

The space-dependent thermal-hydraulic and heat structure variables which are mapped from RELAP5 to PARCS include (in order) moderator temperature (T_m), liquid density (ρ_l), vapor density (ρ_v), void fraction (α), boron concentration (B), average fuel temperature (\bar{T}_f), fuel centerline temperature (T_f^0), and fuel surface temperature (T_f^s). It should be noted that the Doppler temperature to be used in PARCS is determined by the user based on either the average fuel temperature, or a linear combination of the centerline and surface temperatures as shown in Eq. (1).

$$T_{dopl} = (1 - \omega) T_f^0 + \omega T_f^s \quad . \quad (1)$$

For calculating the Doppler temperature based on Eq. (1), the user inputs a single weighting factor (ω) to be used for each node-wise temperature.

The space-dependent total powers obtained from the spatial kinetics calculation, which consist of the sum of fission and decay powers, are mapped from PARCS to both the thermal-hydraulic zones and heat structure components in RELAP5. The amount of energy deposited directly to the thermal-hydraulic zones is based on a user-determined fraction of the total power which is constant for all nodes. Thus, the powers packed into the vector to be sent to the General Interface are (in order) total power to thermal-hydraulic zones ($Q_t^{T/H}$), and total power to heat structure components (Q_t^{HS}).

Each of these variables correspond to a submatrix and a subvector, and the described ordering holds for both the permutation matrices and the vectors. For the permutation matrices, the ordered submatrices form the block structure illustrated in Figure 1. In addition, the dimension of each class of submatrix shown in Figure 1 is consistent with Section IV of the SRS for the General Interface, and is shown below. It is noted that the dimension of each class of subvector is consistent with that of the corresponding submatrix.

- (1) Thermal-Hydraulic to Neutronic Submatrices: number of neutronic nodes by number of thermal-hydraulic zones.
- (2) Heat Structure to Neutronic Submatrices: number of neutronic nodes by number of heat structure components.

- (3) Neutronic to Thermal-Hydraulic Submatrices: number of thermal-hydraulic zones by number of neutronic nodes.
- (4) Neutronic to Heat Structure Submatrices: number of heat structure components by number of neutronic nodes.

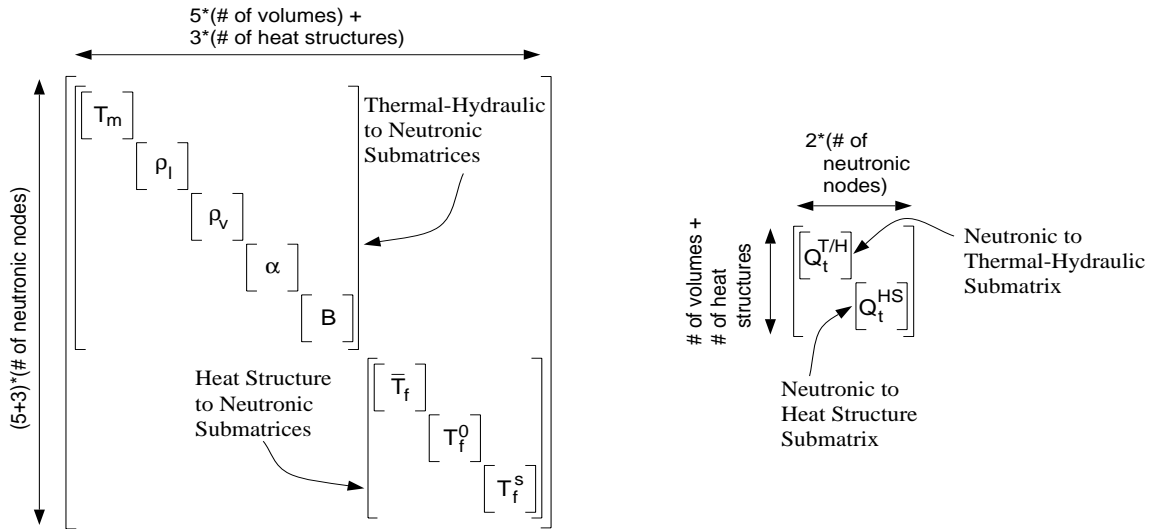


Figure 1: Structure of the Permutation Matrices

It is necessary to ensure that the internal ordering of data in submatrices and corresponding subvectors is consistent. Thus, it is specified that thermal-hydraulic zones are ordered from lowest to highest RELAP5 volume number, heat structures are ordered from lowest to highest RELAP5 component number, and neutronic nodes are ordered from lowest to highest PARCS node number. The RELAP5 volume and component numbers correspond to those numbers specified in the RELAP5 input. The node numbers used by PARCS are computed from PARCS input, and correspond to a plane-wise ordering, as shown in Eq. (2),

$$\text{Node Ordering} = \left[\begin{matrix} [N_1^1 & \dots & N_l^1] \\ \dots \\ [N_1^k & \dots & N_l^k] \end{matrix} \right] ; N_1^1 = 1 , N_l^k = k \times l \quad (2)$$

where l corresponds to the number of nodes in a plane and k corresponds to the number of planes.

II.A.2. Process Control and Information Flow

As mentioned, the PDMR contains three basic functional units, and the communication between the PDMR and the General Interface is achieved with the use of the PVM package. An illustration of the flow of information between RELAP5 and PARCS in each functional unit is shown in Figure 2. The purpose of each unit and the nature of the data to be transferred will be discussed in the following sections.

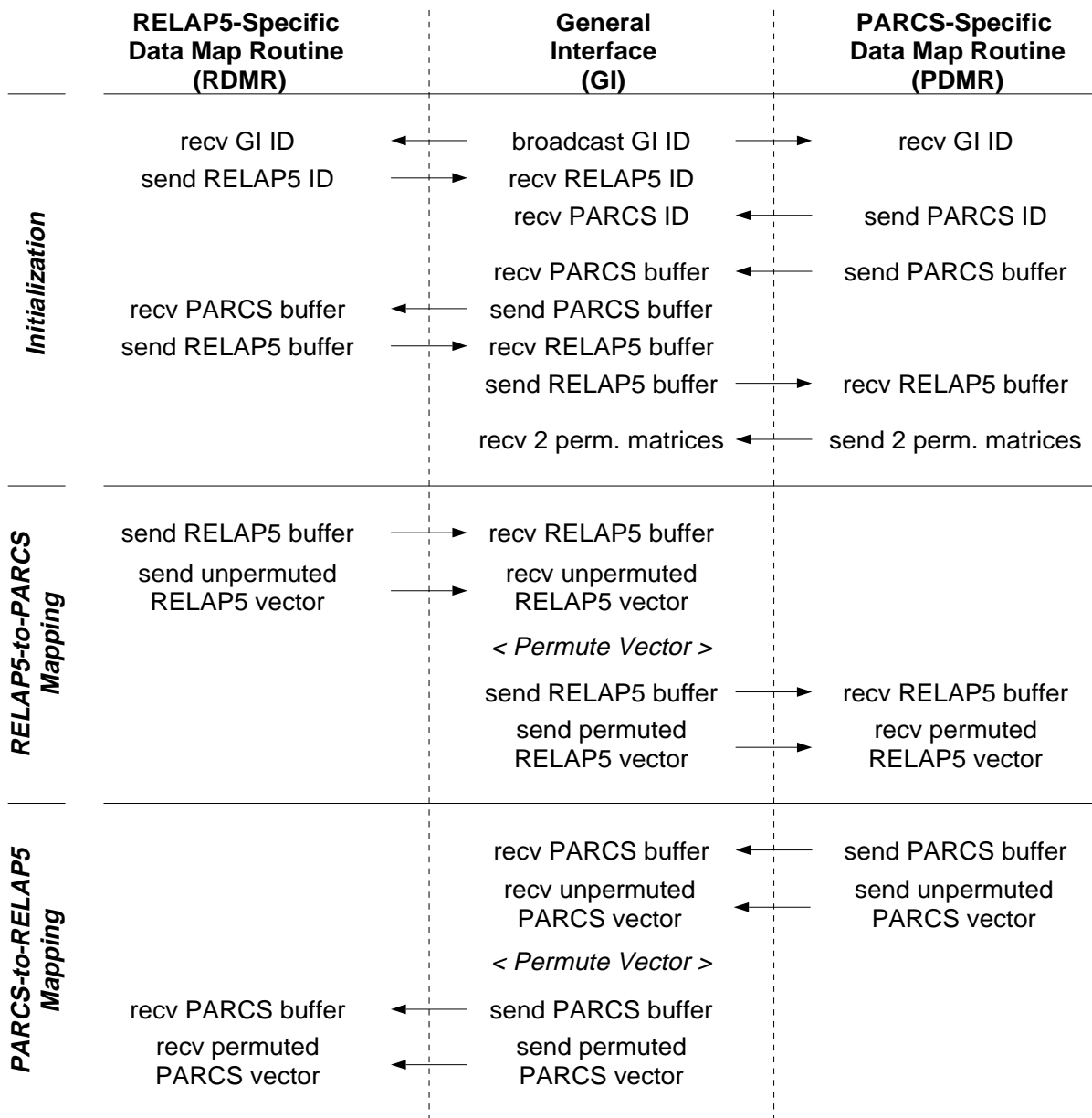


Figure 2: Communication Flow Between RELAP5 and PARCS

The process control for the functional units utilized by this routine resides with PARCS, and is demonstrated in Figure 3. Specifically, following the input processing for PARCS, the initialization unit of the PDMR is called. Upon the return to PARCS, the steady-state eigenvalue calculation is initiated. It should be noted that currently PARCS (Ver. 2.1) does not have a restart capability. The flow logic in Figure 3, which depicts the steady-state eigenvalue problem being bypassed if the calculation is beginning from a restart, will not be implemented. However, this logic is shown to demonstrate the calculational flow for future implementations in which RELAP5 will provide information concerning whether a restart calculation is being performed or

not. For the initial calculation, time is not advanced, and PARCS first calls the PDMR to obtain the thermal-hydraulic and heat structure initial condition data. This data is stored in the PARCS memory, and used to incorporate appropriate feedback into the cross sections. Using these updated cross sections, the linear system is constructed and PARCS performs the eigenvalue calculation, in which the total core power is held constant. PARCS then calls the third unit of the PDMR to send the space-dependent neutronic powers corresponding to the coolant and the fuel to the General Interface.

The time-dependent calculation is initiated by first calling the second unit of the PDMR, which obtains all necessary time- and space-dependent thermal-hydraulic data. Once this data has been received and stored in memory, PARCS incorporates the appropriate feedback into the cross sections, constructs the linear system, and performs the time-dependent spatial kinetics solution for the time step dictated by RELAP5. PARCS then calls the third unit of the PDMR in order to transfer all necessary time- and space-dependent neutronics data to the General Interface. This procedure is repeated until RELAP5 indicates that the calculation should be terminated, or a fault signal is detected.

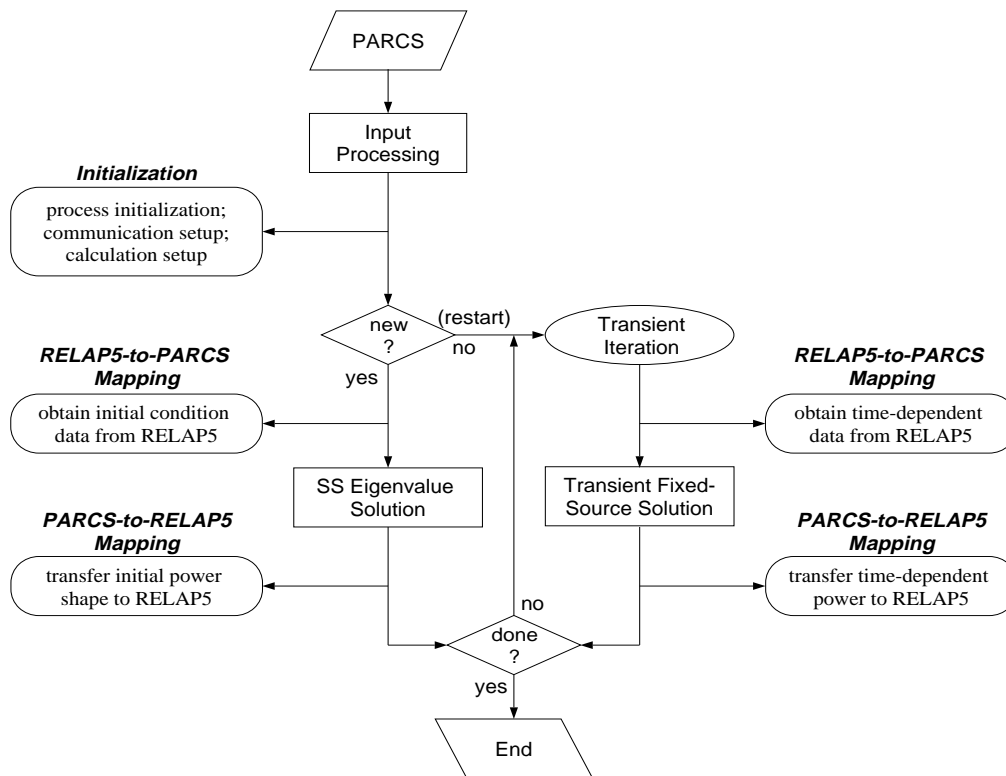


Figure 3: Process Flow in PARCS/PDMR

II.A.3. Initialization

The initialization unit has several functions, the first of which invokes the PVM process for PARCS and sets up the communication with the General Interface process through the transfer of

process IDs. The next task involves processing the user input and the initial control information needed by the RELAP5 process. The input, which is described in detail in Section II.F, consists primarily of the mapping information needed for the two permutation matrices. However, additional input is processed concerning the method of calculating the Doppler temperature to be used in PARCS, as was discussed in Section II.A.1 of this document. The PARCS initial control information, which is sent to the RELAP5 process via the General Interface, is stored in data type-dependent buffers and contains (in order) the following information:

- (1) Indication of an error in PARCS (logical)
- (2) Indication of a data error in the PARCS-Specific Data Map Routine (logical)
- (3) Indication of a PVM error in the PARCS-Specific Data Map Routine (logical)
- (4) Indication of a data error in the General Interface (logical)
- (5) Indication of a PVM error in the General Interface (logical)
- (6) Indication of normal calculation termination (logical)
- (7) Indication of whether PARCS-Specific Data Map Routine is sending the permutation matrices (logical)
- (8) RELAP5 card number corresponding to a control rod trip (integer*4)
- (9) Number of thermal-hydraulic volumes in vessel (integer*4)
- (10) Number of heat structure components in vessel (integer*4)
- (11) Number of columns in thermal-hydraulic to neutronic matrix (integer*4)
- (12) Number of rows in neutronic to thermal-hydraulic matrix (integer*4)
- (13) Array of RELAP5 volume numbers corresponding to the vessel (integer*4)
- (14) Array of RELAP5 heat structure numbers corresponding to the vessel (integer*4)

For items (1) through (5) several modes of failure exist. These modes of failure will be discussed in Section II.A.6 for items (2) and (3). Item (8) is needed by the RDMR to locate the logic in RELAP5 for determining a scram, and is obtained through user input to the PDMR. Items (9) through (14) are needed in the RELAP5-Specific Data Map Routine (RDMR) for mapping vessel property data between RELAP5 and the General Interface, where items (13) and (14) are obtained through user input to the PDMR. It should be noted that item (6) is included only to be consistent with the requirements of the General Interface, and for the initialization unit this will always be "false".

In addition, the initial control information from the RELAP5 process is received via the General Interface, and contains (in order) the following information:

- (1) Array of heat structure geometry descriptors (character*6)
- (2) Indication of an error in RELAP5 (logical)
- (3) Indication of a data error in the RELAP5-Specific Data Map Routine (logical)
- (4) Indication of a PVM error in the RELAP5-Specific Data Map Routine (logical)
- (5) Indication of a data error in the General Interface (logical)
- (6) Indication of a PVM error in the General Interface (logical)
- (7) Indication of normal calculation termination (logical)
- (8) Indication of whether RELAP5-Specific Data Map Routine is sending the permutation matrices (logical)
- (9) Indication of whether this is a "restart" run (logical)

- (10) Time point at which calculation is beginning (real*8)

Item (1) is needed by the PDMMR in the second functional unit to calculate the Doppler temperature. Specifically, the calculation of Doppler temperature based on a combination of centerline and surface temperatures is restricted to heat structures corresponding to a cylindrical fuel pin. Thus, if the user chooses to utilize a combination of centerline and surface temperatures, but RELAP5 indicates that the heat structure is not a pin, then the PDMMR will default to using only the average fuel temperature sent from RELAP5. In future implementations, RELAP5 will control whether the calculation is “new” or based on a “restart”, and thus items (9) and (10) will be needed to ensure that the PARCS calculation is synchronized with the RELAP5 calculation. However, for the current implementation, item (9) will always be “false” and item (10) will always be 0.0 sec.

Following this initial data transfer, the permutation matrices are constructed based on user-input and sent to the General Interface where they are stored for future use.

II.A.4. Data Map from General Interface to PARCS

The requirements of this functional unit are few, and relate only to the receipt of information from the General Interface, and transfer of information to the memory utilized by PARCS. Specifically, time-dependent control information is received from the thermal-hydraulics process via the General Interface. This information is transferred in the same manner as the initial control information previously described, and consists (in order) of the following:

- (1) Indication of an error in RELAP5 (logical)
- (2) Indication of a data error in the RELAP5-Specific Data Map Routine (logical)
- (3) Indication of a PVM error in the RELAP5-Specific Data Map Routine (logical)
- (4) Indication of a data error in the General Interface (logical)
- (5) Indication of a PVM error in the General Interface (logical)
- (6) Indication of normal calculation termination (logical)
- (7) Indication that a restart edit should be performed (logical)
- (8) Indication that a major edit should be performed (logical)
- (9) Indication of a scram (logical)
- (10) Current time step size (real*8)

In future implementations, RELAP5 will control the restart and major output edit frequency, and communicate this information (items (7) and (8)) to PARCS for synchronization purposes. However, in the current implementation, item (7) and item (8) will always be “false”.

In addition, the permuted vector of space-dependent thermal-hydraulic and heat structure data is received from the General Interface and stored in the appropriate memory locations of PARCS for incorporating feedback into the cross sections. In general, no unit conversion is necessary since both PARCS and the General Interface utilize SI units, and since both codes store boron concentration with units of *ppm*. The only exception to this is in the treatment of the moderator and Doppler temperatures, in which PARCS stores the moderator temperature with units of

Celsius, and stores the square root of Doppler temperature with units of \sqrt{K} . Again, the Doppler temperature is calculated from the fuel temperatures received from RELAP5 based on both user specification (i.e. using average fuel temperature or a combination of centerline and surface fuel temperatures) and heat structure geometry, as discussed in the previous sections.

II.A.5. Data Map from PARCS to General Interface

This unit controls the processing and transfer of data to the General Interface. Specifically, the time-dependent control information is packed into data type-dependent buffers and sent to the General Interface. The information contained in these buffers only correspond (in order) to the following error indicators:

- (1) Indication of an error in PARCS (logical)
- (2) Indication of a data error in the PARCS-Specific Data Map Routine (logical)
- (3) Indication of a PVM error in the PARCS-Specific Data Map Routine (logical)
- (4) Indication of a data error in the General Interface (logical)
- (5) Indication of a PVM error in the General Interface (logical)
- (6) Indication of normal calculation termination (logical)

Again, item (6) is included only to be consistent with the requirements of the General Interface. The value in this location is dependent upon the termination signal sent from RELAP5.

Following this, the neutronic space-dependent solution data is mapped into a single vector and transferred to the General Interface to be mapped to the appropriate thermal-hydraulic and heat structure problem domains using the matrix processed during the initialization procedure. Again, no unit conversion is necessary since both PARCS and the General Interface utilize SI units.

II.A.6 Error Checking

The error checking unit is called from each of the three functional units, and is included to help ensure that the permutation matrices and the vectors of property data are consistent with specified requirements. Specifically, the checks performed during the initialization are:

- (1) A neutronic node number specified by user in the matrix input is not available in PARCS.
- (2) A neutronic node exists in PARCS which was not assigned a volume or heat structure number in the matrix input.
- (3) A weighting factor used in the matrix is greater than one or less than zero.
- (4) Eq. (5) of the SRS for the General Interface is not verified.

the checks performed during the RELAP5 to PARCS mapping are:

- (1) Dimension of the permuted vector of RELAP5 data is not equal to the number of rows in the thermal-hydraulic to neutronic matrix.
- (2) Elements of the permuted vector are outside specified bounds. These bounds will be specified in the Software Design and Implementation Document (SDID) for the PDMR.

and the checks performed during the PARCS to RELAP5 mapping are:

- (1) Dimension of the unpermuted vector of PARCS data is not equal to the number of columns in the neutronic to thermal-hydraulic matrix.
- (2) Elements of the unpermuted vector are outside specified bounds. Again, these bounds will be specified in the SDID for the PDMR

In addition to these “data” errors, error checks are performed on all PVM calls in each of the functional units. Specifically, for most of the PVM calls, the final argument is an integer error identifier. If no error has occurred, this value will be greater than or equal to zero. However, if an error is detected, a negative integer, which corresponds to a specific PVM error, will be returned. At this point, the PDMR will print the appropriate error message, and attempt to terminate the other processes as quietly as possible. However, it should be noted that an error occurring in a PVM call might be such that safe termination of the other processes is not possible.

II.A.7 Exit Handling

The termination of the PARCS process is initiated based on either a normal termination signal from RELAP5 or one of several fault signals. If a fault signal is received from RELAP5, the General Interface, or the RELAP5-Specific Data Map Routine, the PDMR simply prints the appropriate error message indicating the origination of error, and begins the exit process. If a fault signal originates in either the PDMR or PARCS, then it is necessary to communicate the appropriate signal, using the PARCS control buffer, to the General Interface process, and subsequently, the RELAP5 process in order to safely terminate the execution. Once this information has been transferred, the PDMR again prints the appropriate error message and begins the exit process.

The exit process, which is the same regardless of whether a normal termination signal or a fault signal is detected, involves first removing PARCS from the PVM process, and then terminating the execution.

II.B. Performance

The performance of the PDMR is largely dependent upon the size of the messages which need to be transferred to and from the General Interface. For accurate modeling of reactor cores, the amount of information needed to couple RELAP5 with PARCS could be substantial, and thus would create a significant overhead in the overall computation time.

Issues of accuracy concerning the PDMR are few, and relate simply to the processing of the permutation matrices from user input during the initialization procedure, and the mapping of data between the memory of PARCS and the buffers used by the General Interface.

II.C. Design Constraints

The design of the PDMR adheres to the FORTRAN-90 standard. Regarding the coupling of PARCS with the General Interface, the process flow for the PDMR should be consistent with that for the General Interface and RELAP5 to ensure communication coherency (e.g. a *send* should have a corresponding *receive*), as was illustrated in Figure 2. In addition, this design should be modular such that implementation requires only minimal modifications to the PARCS source.

In this design, RELAP5 and PARCS are locked into the same time step, and the advancement of the time-dependent calculation is controlled by RELAP5. In future implementations, RELAP5 will also dictate to PARCS when restart and major output edits should be performed based on RELAP5 input. Concerning control rod logic, PARCS treats all "planned" (user-input) rod movement, and RELAP5 controls only the initiation of a scram.

II.D. Attributes

The PDMR requires the use of the PVM package for controlling the communication with the General Interface. Since PVM is portable across all computational platforms, the portability of the PDMR remains unaffected. As mentioned in the SRS for the General Interface, PVM is distributed (and installed, if needed) with the General Interface software in order to reduce the burden on the user.

II.E. External Interfaces

As mentioned, the PDMR is coupled to the General Interface, and consequently to RELAP5, through the use of the PVM package, where messages are communicated utilizing the *send* and *receive* protocols. Thus, process flow through the PDMR is also dependent upon the General Interface. Specifically, the PDMR process is suspended until the appropriate data has been received from the General Interface.

II.F. I/O Requirements

The input requirements for the PDMR are two-fold. First, mapping information is required from the user for the two permutation matrices. Specifically, the mapping information is contained in two tables whose columns are:

Table 1: Thermal-Hydraulic Volume \leftrightarrow Neutronic Node Mapping

RELAP5 Volume Number,	PARCS Node Number,	Weighting Factor
...

Table 2: Heat Structure Component \leftrightarrow Neutronic Node Mapping

RELAP5 Component Number,	PARCS Node Number,	Weighting Factor
...

where no restrictions are placed on the user concerning the order of the mapping data. Two additional user input items, which are not required, are also processed. First, user input is needed concerning the method of calculating Doppler temperature. If the user chooses to utilize a combination of centerline and surface temperatures, then the weighting factor shown in Eq. (1) is also required as input. If the Doppler input option is not provided, the default will be to use the linear combination, where the weighting factor is set to 0.7. Second, RELAP5 does not know a priori what trip unit number corresponds to a control rod scram. Thus, the user should input the control rod trip unit number specified in the RELAP5 input. This input is not required if the RELAP5 model does not have a trip card corresponding to a control rod scram. Sufficient error checking will be provided to ensure that the data is input by the user in the correct form.

The second type of input relates to the data received from the General Interface, and consists of RELAP5 control information and the vector of thermal-hydraulic property data. The output of the PDMR relates to the error/warning messages which are written to either the screen or the disk, and to the information which is communicated with the General Interface. The latter is composed simply of the permutation matrices, the vector of space-dependent neutronic data, and the PARCS control information.

The only additional input requirement for PARCS is a user option indicating that an external system thermal-hydraulics code should be utilized. If this indication is made, the PDMR is activated. If this option is not utilized, PARCS performs the calculation using its internal thermal-hydraulic routines.

III. Summary

This document has described the software design requirements necessary to successfully couple PARCS with the General Interface and RELAP5. The key features of this design include the utilization of the PVM software to control communication, and the modular implementation of this subroutine to minimize the modification necessary to PARCS. In addition, the use of the PDMR as a secondary interface allows the design of the General Interface to remain unchanged.

IV. References

1. D. Barber and T. Downar, "Software Requirements Specification for the General Interface in the Coupled Code," Technical Report, PU/NE-98-8, Purdue University, (1998).
2. W. Wang, G. Mortensen, D. Barber, and T. Downar, "Software Requirements Specification for the RELAP5-Specific Data Map Routine in the Coupled RELAP5/PARCS Code," Technical Report, SCIE-NRC-368-98, SCIENTECH, Inc., (1998).
3. Y. Saad, "Numerical Methods for Large Eigenvalue Problems," Manchester University Press, Manchester, UK., pp. 40-41 (1992).