The Structure of an Interface Linking NASCAD and RIM

William J. Rasdorf, Associate Professor
Osama Y. Abudayyeh, Research Associate
Sivand Lakmazaheri, Research Associate
Civil Engineering Department
North Carolina State University
Raleigh, NC 27695

1 Abstract

Data transfer between different application programs is an important and crucial issue to consider if a successful integration is to be accomplished between the different program modules of a large computing system. Two system modules are said to be interfaced if there exists a data path between them. To achieve such data paths between a number of heterogeneous application programs, interface programs need to be developed to map the inputs and outputs of modules to and from a database.

This paper discusses the NASCAD–RIM Interface program version 1.0, that was developed to provide a link between the NASCAD geometric modeller and the RIM relational database management system. These two software systems are parts of an overall CAD/CAM system for the design and manufacturing of custom orthopedic footwear. The NASCAD-RIM Interface emphasizes the importance of having a centralized database management system as the core of an integrated CAD/CAM system.

A brief description of the two application software packages for which the interface program was developed (NASCAD and RIM) is given. An overview of the interface system that provides the communication path between NASCAD and RIM is also presented. The paper then discusses the interface system in terms of its components, the NASCAD-TO-RIM and RIM-TO-NASCAD interfaces, and it also discusses the RIM and NASCAD databases. Examples that demonstrate how the interface is used and what it does are also provided.

2 Introduction and Background

The NASCAD–RIM INTERFACE system was developed as a part of a project entitled "Engineering Specification and System Design for CAD/CAM of Custom Shoes–Data Representation and System Integration." The purpose of this project is to develop an integrated CAD/CAM system for the design and manufacture of custom orthopedic footwear. The architecture of the overall integrated system was presented at the 1988 ASME Computers in Engineering Conference [Rasdorf88a]. This paper continues the description of the evolving system by focusing on two of its components (NASCAD and RIM) and the interaction between them.

What is to be described herein is a mechanism for transferring data between the geometric modeller NASCAD (NASA Computer Aided Design) and the centralized relational database management system RIM (Relational Information Manager) so that other application programs can

share geometric data with NASCAD via the database. A secondary effect of the interface system is the availability of a means for modifying graphical models created by NASCAD using a database management system.

This section provides an introduction to the two application software packages for which the interface program was developed, that is, NASCAD and RIM. The interface program was written to provide a link between those two software packages in both directions, that is, from NASCAD to RIM and from RIM to NASCAD. For additional details on either NASCAD or RIM refer to their users manuals, [Cosmic86a] and [Gray82].

2.1 NASCAD

NASCAD is the geometric modeller of the software package NEXUS (NASA Engineering eXtendible Unified Software). NEXUS is an integrated set of computer programs whose function is to support the activities encountered in many engineering projects. NASCAD enables the user to construct complex models of physical objects using primitives such as points, lines, and polygons.

The NASCAD user language NEIL (Nascad Extendible Interactive Language) provides capabilities that include procedural definition, conditional execution, arithmetic and character operations, database manipulation, and graphics operations. These capabilities can be utilized to define macros for facilitating definition and manipulation of geometric models.

The structure of the NASCAD database is a tri-level hierarchy consisting of "elements," "components," and "entities," [Cosmic86]. In NASCAD each element is composed of one or more components and each component contains two or more entities. Physically, the NASCAD database forms a double ring structure is as shown in Figure 1. The nodes in the primary ring constitute the elements and the nodes in the secondary ring constitute the components of NASCAD. Attached to each component is a list of its constituent entities.

NASCAD supports four types of elements:

1. Cell Definitions: Contain the NASCAD graphics data,
2. Macros: User defined commands for interacting with NASCAD,
3. Variables: User defined symbols for calculations, and
4. Relational Tables: Non-graphics relational data tables.

There are eleven types of components in NASCAD as follows:

1. Polyarc: A combination of points and arcs,
2. Text: A component consisting of text only,
3. Polygon: A closed polyarc,
4. Stretchable Rectangle: A rectangle capable of generating cross-hatch at display time,
5. Patch: A polygon with a surface normal,
6. Bicubic Spline Surface: A surface consisting of four spline curves,

7. Ruled Surface: A surface consisting of two polyarcs,
8. Drive Surface: A surface generated by moving one curve down another,
9. Drive Volume/Solid: A drive surface with end caps,
10. Spline Component: A spline curve consisting of two spline entities, and
11. Cell Instance: Pointer to a cell definition.

Entities are the basic building blocks of the NASCAD components. The five types of entities currently being supported by NASCAD are:

1. Point: A point with three coordinates,
2. Arc: Used for defining circles, arcs, ellipses, hyperbolas, and parabolas,
3. Text: Contains information about the aspect ratio, slant, justification, and path of text components,
4. Cell Instance: Contains offset and rotational information about the Cell Instance component, and
5. Spline Entity: Consists of a position vector and a tangent vector.

The NASCAD database (the double ring structure) can be stored in a file. This file is then referred to as a NASCAD data file.

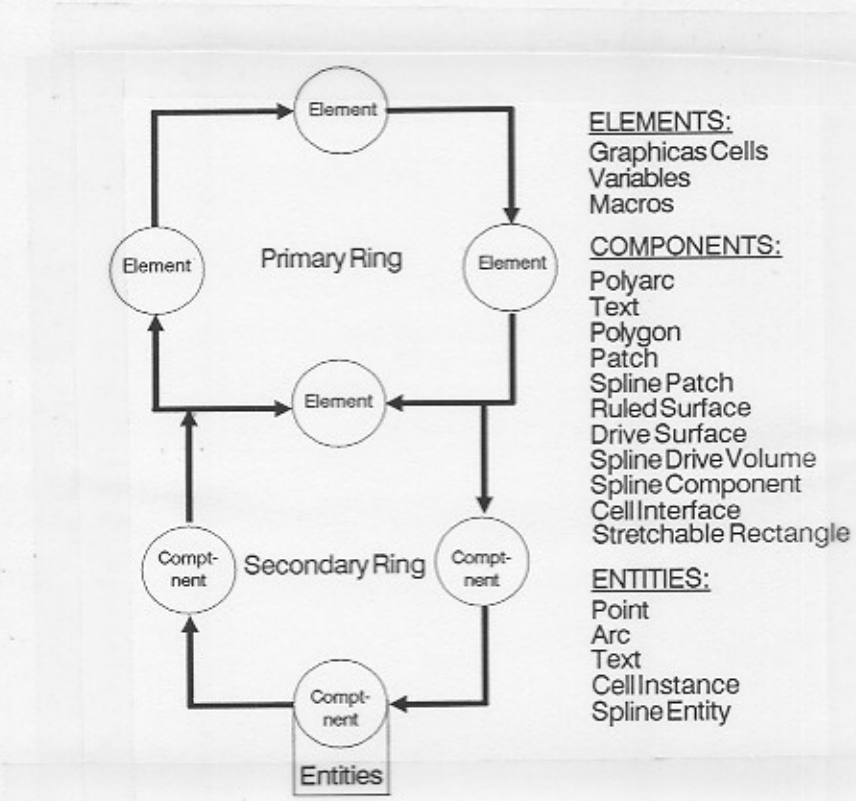

Figure 1: NASCAD Data Structure

2.2 RIM

RIM is a relational database management system developed by Boeing Commercial Aircraft Company under the auspices of the IPAD (Integrated Programs for Aerospace Vehicle Design) Project Office of NASA's Langley Research Center. RIM is suitable for most scientific applications since it supports scalar, vector, and matrix attributes of types integer, real, and double precision. Furthermore, RIM enables one to define vectors of fixed or variable length. RIM can be invoked as a stand-alone system or through an application program interface. The RIM application program interface contains a library of FORTRAN modules which permit access to a database interactively created with the RIM database management system [Gray82].

RIM has been used as the central database management system of many computer based analysis systems at the NASA Langley Research Center and elsewhere. Using RIM as the relational database management system in the footwear CAD/CAM project allowed for further transfer of this NASA developed technology to industry and made available to the project the large body of RIM expertise at NASA.

3 Overall System Integration

The role of RIM as a centralized database management system in an overall integrated system is twofold. On one hand RIM is used as a data transfer mechanism. This function of RIM facilitates the transfer of data between a set of heterogeneous application programs and enables each application program to access the data independent of its physical organization in the database. As a data transfer mechanism, RIM assumes a dynamic role in the evolving design process and provides the data required by each application program throughout the design process.

On the other hand, RIM is also used as a passive data repository. This capability provides the overall system with a centralized and manageable storage area for capturing the information obtained at different stages of design from each application program. The data integrity of the process can thus more easily be maintained as can a historical record of the evolving design [Rasdorf87].

To transfer data between independent application programs one needs to interface the corresponding application programs. In an integrated system two components are said to be interfaced if there exists a data path between the two components. In other words, an interface program maps the output data transmitted from one program to the input data needed by another program. Thus, our task was initially that of defining mapping functions between different application programs. Each individual mapping function maps the data as perceived by its corresponding application program to a set of appropriate database relations in RIM. The collection of all the relations in RIM represents the data transmitted between the application programs throughout the design process.

In order to interface the NASCAD application program with RIM and to support the NASCAD graphics data in RIM, eight relations have been provided. These relations are designed for supporting all of the NASCAD elements, components, and entities. A detailed definition of these relations is given in Section 4.2.

Conceptually, the NASCAD–RIM Interface system provides a mapping function between NASCAD and RIM, two of the many applications in the overall Custom Footwear Design System. This mapping function enables one to completely map the NASCAD graphics data to its corresponding representation in RIM and vice versa. The NASCAD–RIM Interface system, shown in Figure 2, is

composed of NASCAD-TO-RIM and RIM-TO-NASCAD interfaces. These interfaces are responsible for transferring the data from a NASCAD data file to a RIM database and from a RIM database to a NASCAD data file, respectively.

4 NASCAD–RIM Interface System

The section discusses the interface system in terms of its components, that is, the NASCAD-TO-RIM interface and the RIM-TO-NASCAD interface. A description of the interface program structure is also provided. The section is divided into three subsections: Processes, Data, and NASCAD–RIM Interface Program Structure. The Processes subsection discusses the two component interfaces and their data transformation strategy; the Data subsection discusses the RIM database; and the NASCAD–RIM Interface Program Structure subsection discusses the structure of the FORTRAN code of the interface system.

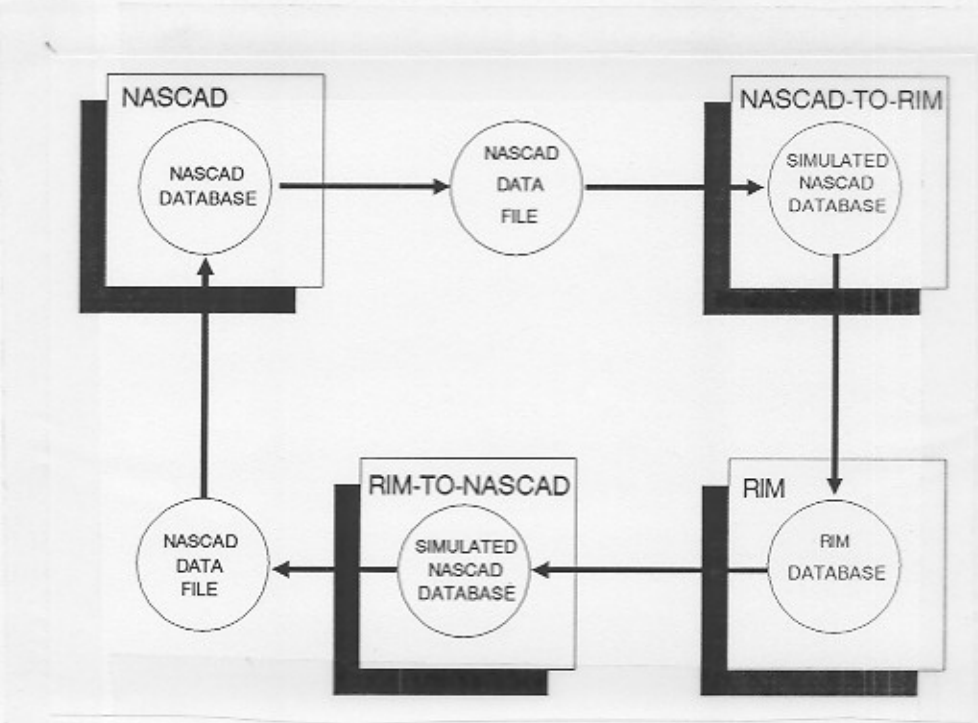

Figure 2: NASCAD–RIM Interface System

4.1 Processes

4.1.1 NASCAD-TO-RIM Subsystem

This subsystem simulates the structure of the NASCAD global database (a circular linked list), loads data from the NASCAD data file into the simulated database, extracts the data from the simulated database, and stores it in the corresponding relations in the RIM database. This subsystem is organized into two sets of modules. One set of modules, FROM-NASCAD, retrieves data from the

NASCAD database. The other set of modules, TORIM, stores the data in the RIM database. The NASCAD data structure allows the modularization of the data extraction process by implementing a number of submodules for retrieving the different data types supported by NASCAD.

The FROM-NASCAD module contains three submodules named A, B, and C. These submodules extract the data associated with NASCAD "elements," "components," and "entities," respectively. Module C contains a set of submodules (C1 through C5) which extract data associated with the NASCAD entities POINT, ARC, TEXT, CELL-INSTANCE, and SPLINE, respectively.

The TORIM module is composed of eight submodules. These submodules, named S, T, U, V, W, X, Y, and Z, store the data extracted from NASCAD in the database relations ELEMENTS, COMPONENT, ENTITIES, POINT, ARC, TEXT, CELL-INSTANCE, and SPLINE, respectively. Figure 3 illustrates the NASCAD-TO-RIM subsystem.

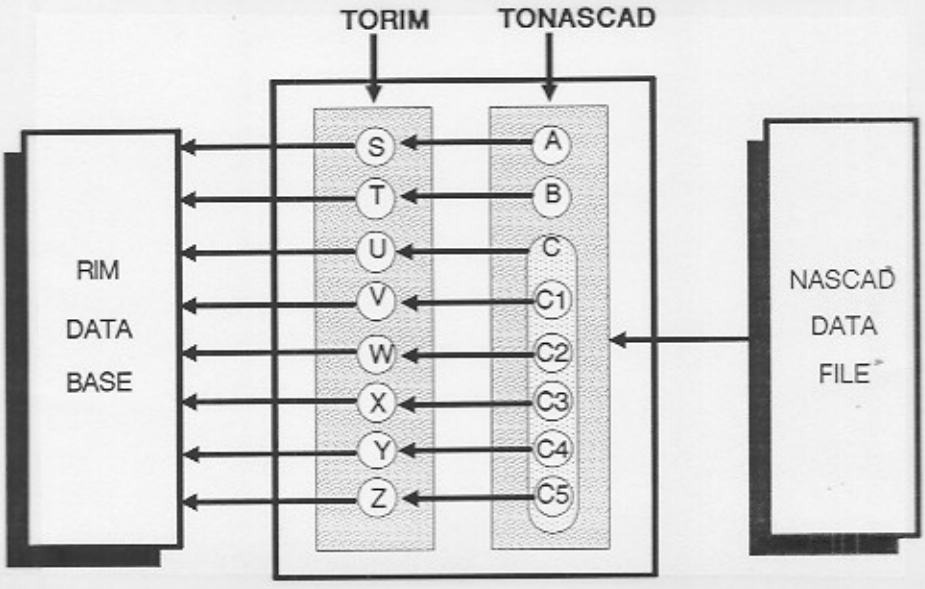

Figure 3: NASCAD-TO-RIM Subsystem

4.1.2 RIM-TO-NASCAD Subsystem

This subsystem simulates the structure of the NASCAD global database (a circular linked list), extracts data from RIM database, converts the data into a form recognizable by NASCAD and stores it in the NASCAD simulated database. When all the RIM data is transferred to the NASCAD simulated database, the simulated database is stored in the NASCAD data file. The logical organization of this subsystem is very similar to that of the NASCAD-TO-RIM subsystem. In this subsystem, a set of FROM-RIM modules extracts data from RIM and a set of TONASCAD modules stores the extracted data in the NASCAD database in a manner similar to that described in the preceeding section.

4.2 Data

This section contains a description of the relational database created in RIM to support the NASCAD graphics data [Rasdorf88a]. The RIM database consists of eight low level relations corresponding to NASCAD's elements, components, and entities. They have been drawn from the NASCAD database structure but they are not exactly the same. The following is a description of these relations, where the key attributes are underlined.

- ELEMENT(ElementId,ElementType,ElementName, ComponentVector)
  This relation contains name, type, and component IDs of the NASCAD elements.
  - ElementId: A unique identifier for relation ELEMENT.
  - ElementType: Type of element.
  - ElementName: Name of element.
  - ComponentVector: The vector containing the element's component IDs.
- COMPONENT(ComponentId,ComponentName,ComponentType, ComponentGraphicsVector,EntityVector,E

  This relation contains name, type, graphics information, and entity IDs of the NASCAD components.
  - ComponentId: A unique identifier for relation COMPONENT.
  - ComponentName: Name of component.
  - ComponentType: Type of component.
  - ComponentGraphicsVector: Vector of the component's graphics information.
  - EntityVector: Vector containing the component's entity IDs.
  - ElementId: A unique identifier for relation ELEMENT.
- ENTITIES(EntityId,EntityType)
  This relation contains information about type of the NASCAD entities.
  - EntityId: A unique entity identifier.
  - EntityType: Type of entity.
- POINT(EntityId,PointXYZ,PointGraphicsVector, ComponentId)
  This relation contains information about the POINT entities generated in NASCAD.
  - EntityId: A unique entity identifier.
  - PointXYZ: X,Y, and Z coordinates of the point.
  - PointGraphicsVector: Vector of the point graphics information.
  - ComponentId: A unique identifier for relation COMPONENT.
- ARC(EntityId,ArcType,StartPoint,CenterPoint,A-axis, B-axis, ArcGraphicsVector,ComponentId) This relation contains information about the ARC entities generated in NASCAD.
  - EntityId: A unique entity identifier.
  - ArcType: Type of arc.
  - StartPoint: X, Y, and Z coordinates of the arc starting point.
  - CenterPoint: X, Y, and Z coordinates of the arc center point.

- A-axis: The major axis difference vector.
- B-axis: The minor axis difference vector.
- ArcGraphicsVector: Vector of the arc graphics information.
- ComponentId: A unique identifier for relation COMPONENT.

- TEXT(EntityId,Origin,Aspect,Slant,Path,Offset, TextGraphicsVector,TextString,ComponentId) This relation contains information about the TEXT entities generated in NASCAD.
  - EntityId: A unique entity identifier.
  - Origin: X, Y, and Z coordinates of the text origin.
  - Aspect: Text aspect ratio.
  - Slant: Text slant.
  - Path: Text path.
  - Offset: Coordinates of the text starting point offset.
  - TextGraphicsVector: A vector containing the text graphics information.
  - TextString: A sequence of characters representing the text.
  - ComponentId: A unique identifier for relation COMPONENT.

- CELL(EntityId, CellOrigin,A-axis,B-axis,ComponentId)
  This relation contains information about the CELL INSTANCE entities generated in NASCAD.
  - EntityId: A unique entity identifier.
  - CellOrigin: X, Y, and Z coordinates of the cell origin.
  - A-axis: The major axis difference vector.
  - B-axis: The minor axis difference vector.
  - ComponentId: A unique identifier for relation COMPONENT.

- SPLINE(EntityId,SplineGraphicsVector, PositionVector,TangentVector,ComponentId)
  This relation contains information about the SPLINE entities generated in NASCAD.
  - EntityId: A unique entity identifier.
  - SplineGraphicsVector: Vector of the spline graphics information.
  - PositionVector: Spline position vector.
  - TangentVector: Spline tangent vector.
  - ComponentId: A unique identifier for relation COMPONENT.

4.3 NASCAD–RIM Interface Program Structure

The NASCAD–RIM Interface system is composed of two separate programs: TORIM and TONASCAD [Rasdorf88b]. The TORIM program transforms a graphical model created in NASCAD into a relational data model and stores it in RIM. The TONASCAD program, on the other hand, transfers the data model from RIM back to NASCAD as described in Section 4.1.

Figure 4 shows the hierarchical structure of the TORIM program. The TORIM program takes advantage of some of the existing NASCAD and RIM routines to achieve the data transfer task between the NASCAD and RIM environments. These routines are labeled "NASCAD routine" and "RIM routine" in the figure. Figure 5 shows the hierarchical structure of the TONASCAD

program. The TONASCAD program also takes advantage of existing NASCAD and RIM routines. These routines are labeled as before in Figure 5. Both programs are described below.

The TORIM interface simulates the internal NASCAD database by invoking the AASYSBGN routine. The SETISUB, INFILENAME, and FILEIO are then invoked to set the I/O channel to the input mode, define the NASCAD filename, and load the data into the simulated NASCAD database, respectively. Next, the OPEN$RIM module is invoked to open the RIM database and establish pointers to the appropriate RIM relations via the RMOPEN and RMFIND routines, respectively. The TRAVELEM routine is invoked for traversing the circular linked list that represents the NASCAD database structure. Each node of this linked list is a NASCAD element, and it too has a structure of its own - a circular linked list. A node of this linked list, however, represents a NASCAD component. The routine TRAVCMP processes all the NASCAD components by traversing this component linked list structure.

Each component of NASCAD is composed of a set of NASCAD entities. These entities are retrieved using the TRAVENT routine which determines the type of each entity and invokes the appropriate module for storing the entity data in the RIM database. There are five such modules, namely the POINT$ENTITY, ARC$ENTITY, TEXT$ENTITY, CELL$ENTITY, and SPLINE$ENTITY routines. These routines retrieve the point, arc, text, cell, and spline entities from the NASCAD database and store them in the POINT, ARC, TEXT, CELL, and SPLINE RIM relations, respectively. Then, the appropriate information is placed in the RIM relations ENTITIES, COMPNENT, and ELEMENT by invoking the PUTRIM_ENT, PUTRIM_COMP, and PUT$RIM ELEMENT routines, respectively. These routines use RMFIND and RMLOAD routines to establish a pointer to the appropriate RIM relation and to load the relation with data, respectively. Finally, the RIM database is closed by calling the RMCLOS routine.

The TONASCAD routine simulates the NASCAD internal database by calling AASYSBGN, opens the RIM database by calling OPEN$RIM, and transfers the data from RIM to NASCAD as follows. A NASCAD graphics cell is generated for each row of the relation ELEMENT. This is done by identifying the components associated with each cell, constructing the data structure for the cell, and transferring the data from the RIM database to the graphics cell. The ELEMENT routine identifies the cell components. The DEFCELLN routine constructs the cell data structure and the routine LEVEL2 transfers the data from the RIM database to the NASCAD database. The data transfer is accomplished by retrieving all the components of the current graphics cell from the RIM database and storing them in the NASCAD database. The routines RMFIND, RMWHER, and RMGET establish a pointer to the relation COMPNENT, select the appropriate entities, and retrieve the entity identifiers from the RIM database, respectively. The COMPNENT and LEVEL3 routines extract the entity data from the RIM database and store them in NASCAD database. For each entity, LEVEL3 identifies the RIM relation that holds the entity data by calling the ENTITIES subroutine. Then, one of the POINT, ARC, STRING, CELL, or SPLINE routines is invoked to extract the data from the POINT, ARC, TEXT, or CELL relations, respectively, and store them in the NASCAD database. The actual data extraction is done via the RMFIND, RMWHER, and RMGET routines. Finally, the NASCAD internal database is written to a file by calling the STORE routine. This routine sets the I/O channel to output mode, identifies the output filename, and transfers the data from the internal database to external file. These last three tasks are done by calling the SETISUBI, OUT FILE, and FILEIO routines, respectively.

```
TORIM
    AASYSBGN (NASCAD routine)
    SETISUB
    INFILENAME
    FILEIO (NASCAD routine)
    OPEN$RIM
        RMOPEN (RIM routine)
        RMFIND (RIM routine)
    PROCESS
        TRAVELEM
            TRAVCMP
                TRAVENT
                    POINT$ENTITY
                        RMLOAD (RIM routine)
                    ARC$ENTITY
                        RMLOAD (RIM routine)
                    TEXT$ENTITY
                        RMFIND (RIM routine)
                        RMLOAD (RIM routine)
                    CELL$ENTITY
                        RMFIND (RIM routine)
                        RMLOAD (RIM routine)
                    SPLINE$ENTITY
                        RMFIND (RIM routine)
                        RMLOAD (RIM routine)
                    PUT$RIM_ENT
                        RMLOAD (RIM routine)
                PUT$RIM_COMP
                    RMFIND (RIM routine)
                    RMLOAD (RIM routine)
            PUT$RIM_ELEMENT
                RMFIND (RIM routine)
                RMLOAD (RIM routine)
    RMCLOS (RIM routine)
```

Figure 4: Hierarchical Structure of the TORIM Program

```
TONASCAD
    AASYSBGN (NASCAD routine)
    OPEN$RIM
        RMOPEN (RIM routine)
        RMFIND (RIM routine)
    ELEMENT
    DEFCELLN
    LEVEL2
        RMFIND, RMWHER, RMGET (RIM routines)
        COMPONENT
        LEVEL3
            ENTITIES
            POINT
                RMFIND, RMWHER, RMGET (RIM routines)
            ARC
                RMFIND, RMWHER, RMGET (RIM routines)
            STRING
                RMFIND, RMWHER, RMGET (RIM routines)
            CELL
                RMFIND, RMWHER, RMGET (RIM routines)
            SPLINE
                RMFIND, RMWHER, RMGET (RIM routines)
    STORE
        SETISUB1
        OUT_FILE
        FILEIO (NASCAD routine)
```

Figure 5: Hierarchical Structure of the TONASCAD Program

5 Running the Program

This section demonstrates the use and operation of the interface program by running a complete example. The example is simple, intended only to demonstrate the functionality of the interface. The example starts by creating a model on NASCAD, transferring the model to RIM by running the TORIM portion of the interface, modifying the model using the RIM query language, transferring the modified model back to NASCAD by running the TONASCAD portion of the interface, and displaying the modified model. The example tests the BICUBIC SPLINE SURFACE and PATCH components of NASCAD. A listing of the command file used to create and display the model, in batch mode, before transferring it to RIM can be found in the NASCAD–RIM Interface user's manual [Rasdorf88b].

In RIM, the model was modified using the RIM query language. The following command was issued to modify the position of a set of points:

R> CHANGE POSITION TO (0.5,0.,10.) IN SPLINE WHERE POSITION EQ (1.,0.,10.)

Essentially this command modified the coordinates of a point on the front face of the cylinder in Figure 6 to produce Figure 7. The resulting changes in the SPLINE relation due to this command are shown below.

RELATION : SPLINE

ENTID	SGVEC	POSITION			TANGENT			COMPID
55	0	.707	.707	0.	0.	0.	0.	50
62	0	.707	.707	10.	0.	0.	0.	50
69	0	1.	0.	0.	0.	0.	0.	50
76	0	.5	0.	10.	0.	0.	0.	50
83	0	.707	.707	0.	1.	-1.	0.	50
90	0	1.	0.	0.	0.	-1.	0.	50
97	0	.707	.707	10.	1.	-1.	0.	50
104	0	.5	0.	10.	0.	-1.	0.	50
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
254	0	.5	0.	10.	0.	0.	0.	242
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
289	0	.5	0.	10.	0.	-1.	0.	242
.	.	.	.	.	.	.	.	.

This modified model was then transferred back to NASCAD by issuing the command:

$ RUN TONASCAD

The program responded by asking the user to enter the RIM database name to be transferred and the name of the file in NASCAD into which it was transferred. Figure 7 shows the final modifications to the model that were made while it was stored in RIM.

For a more detailed description on how to use the system, including additional examples, refer to the user's manual [Rasdorf88b].

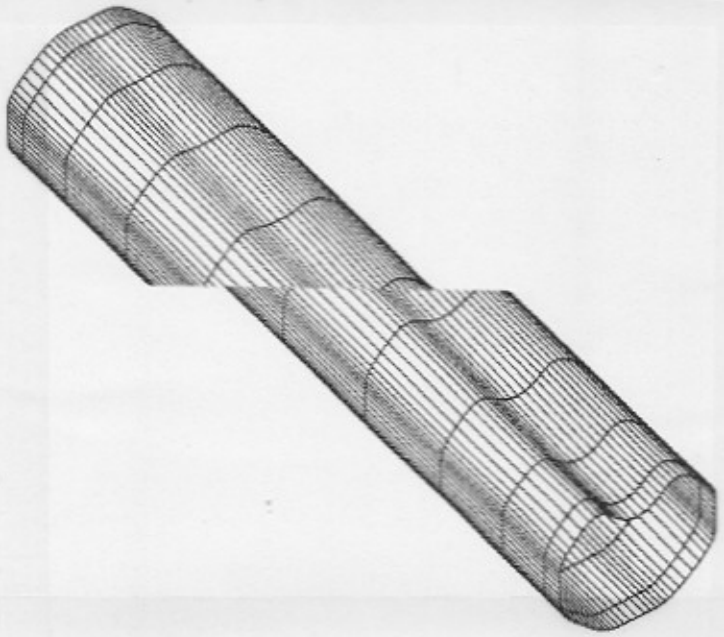

Figure 7: The Model After Modifying It in RIM and Transferring It Back to NASCAD

6 Summary

This paper focused on the description of the NASCAD–RIM Interface system that was developed to provide a mechanism for transferring data between the NASCAD geometric modeller and the centralized RIM relational database management system so that other application programs can share data with NASCAD via the database. An overview of both NASCAD and RIM was provided. The paper described in detail the structure of the interface system in terms of its components, that is, the NASCAD–TO–RIM interface and the RIM–TO–NASCAD interface, and it also described the RIM and NASCAD databases. Finally, an example was presented to demonstrate the use and operation of the interface system.

Both NASCAD and RIM are widely used programs developed by NASA. Both are currently available from the NASA's Computer Software Management and Information Center [Cosmic86b]. The software developed at NCSU provides a valuable link between these two programs. It is the intent of the researchers at NCSU to make this interface software available to the community of users who may have an interest in using it. The software is written in standard FORTRAN 77 and is, therefore, highly portable. Further information about the system, including access to the source code and user's manual, can be obtained from the first author.

7 Acknowledgment

This work was sponsored by the National Aeronautics and Space Administration under Grant NAG-1-883 entitled "Engineering Specification and System Design for CAD/CAM of Custom Shoes–Data Representation and System Integration." The support of this organization is gratefully acknowledged.

References

[Gray82] Gray, F. and Wahlstrom, S., "User Guide: RIM 5.0," Boeing Commercial Airplane Company, 1982.

[Cosmic86a] "NEXUS/NASCAD System Handout," Cosmic Software Services, October 1986.

[Cosmic86b] "Software Catalog," Computer Software Management and Information Center (COSMIC), Athens, GA, 1986.

[Rasdorf88a] Rasdorf, W.J., Lakmazaheri, S., Boose, L.K., and Wallace, R.J., " The Architecture of an Integrated CAD/CAM System for the Design and Manufacturing of Custom Orthopedic Footwear," Proceedings of the Symposium on Engineering Database Management: Critical Issues, American Society of Mechanical Engineers, San Francisco, CA, August 1988. Also in Managing Engineering Data: The Competitive Edge, R. Fulton, Editor, American Society of Mechanical Engineers, San Francisco, CA, 1988.

[Rasdorf88b] Rasdorf, W.J., Lakmazaheri, S., and Abudayyeh, O.Y., "NASCAD–RIM Interface User's Manual," Civil Engineering Department, North Carolina State University, Raleigh, NC, July 1988.

[Rasdorf87] Rasdorf, W.J., "Database Management System Extensions for Engineering Applications," Computers in Mechanical Engineering (CIME), American Society of Mechanical Engineers, Volume 5, Number 5, Pages 62–69, March 1987.

Appendix A: The RIM Database Relations for the Model

This appendix lists the schema and a portion of the data as it exists in the RIM database after creating the model in NASCAD. The data was retrieved by first using the following RIM command:

R> OPEN NASCAD

where NASCAD is the RIM database filename given for the model. Then, the schema and its inherent specifications of all the relations available in that database where listed using the following RIM command:

R> LISTREL ALL

Finally, the contents of each relation were displayed using the following RIM command:

R> SELECT ALL FROM *relation name*

where *relation name* varies according to the desired relation.

The following pages first list the relations and their attributes as they appear in the database in response to the LISTREL command. Next, a portion of the contents of some of the relations is given using the SELECT command. The reader may refer to Figure 1 while reviewing the data and note the connectivity between the relations, particularly with respect to the IDs.

RELATION : ELEMENT

LAST MOD : 89/01/10 READ PASSWORD : NONE
SCHEMA : NASCAD MODIFY PASSWORD : NONE

NAME	TYPE	LENGTH	KEY
ELEMID	INT	1	YES
ELEMTYPE	TEXT	8 CHARACTERS	
ELEMNAME	TEXT	8 CHARACTERS	
COMPVEC	IVEC	VARIABLE	

CURRENT NUMBER OF ROWS = 2

RELATION : COMPNENT

LAST MOD : 89/01/10 READ PASSWORD : NONE
SCHEMA : NASCAD MODIFY PASSWORD : NONE

NAME	TYPE	LENGTH	KEY
COMPID	INT	1	YES
COMPNAME	TEXT	32 CHARACTERS	
COMPTYPE	TEXT	32 CHARACTERS	
CGVEC	IVEC	8	
ENTVEC	IVEC	VARIABLE	
ELEMID	INT	1	YES

CURRENT NUMBER OF ROWS = 9

RELATION : ENTITIES

LAST MOD : 89/01/10 READ PASSWORD : NONE
SCHEMA : NASCAD MODIFY PASSWORD : NONE

NAME	TYPE	LENGTH	KEY
ENTID	INT	1	YES
ENTTYPE	TEXT	8 CHARACTERS	

CURRENT NUMBER OF ROWS = 65

RELATION : POINT

LAST MOD : 89/01/10 READ PASSWORD : NONE
SCHEMA : NASCAD MODIFY PASSWORD : NONE

NAME	TYPE	LENGTH	KEY
ENTID	INT	1	YES
POINTXYZ	DVEC	3	
PGVEC	IVEC	3	
COMPID	INT	1	YES

CURRENT NUMBER OF ROWS = 0

RELATION : ARC

LAST MOD : 89/01/10 READ PASSWORD : NONE
SCHEMA : NASCAD MODIFY PASSWORD : NONE

NAME	TYPE	LENGTH	KEY
ENTID	INT	1	YES
ARCTYPE	TEXT	8 CHARACTERS	
STARTXYZ	DVEC	3	
CENTRXYZ	DVEC	3	
A_AXIS	DVEC	3	
B_AXIS	DVEC	3	
AGVEC	IVEC	2	
COMPID	INT	1	YES

CURRENT NUMBER OF ROWS = 0

RELATION : TEXT

LAST MOD : 89/01/10 READ PASSWORD : NONE
SCHEMA : NASCAD MODIFY PASSWORD : NONE

NAME	TYPE	LENGTH	KEY
ENTID	INT	1	YES
ORIGIN	DVEC	3	
ASPECT	DVEC	3	
SLANT	DVEC	3	
PATH	RVEC	2	
OFFSET	DVEC	3	
TGVEC	IVEC	9	
STRING	TEXT	VARIABLE	
COMPID	INT	1	YES

CURRENT NUMBER OF ROWS = 0

RELATION : CELL

LAST MOD : 89/01/10 READ PASSWORD : NONE
SCHEMA : NASCAD MODIFY PASSWORD : NONE

NAME	TYPE	LENGTH	KEY
ENTID	INT	1	YES
CELLXYZ	DVEC	3	
A_AXIS	DVEC	3	
B_AXIS	DVEC	3	
CELLGVEC	IVEC	2	
COMPID	INT	1	YES

CURRENT NUMBER OF ROWS = 1

RELATION : SPLINE

LAST MOD : 89/01/10 READ PASSWORD : NONE
SCHEMA : NASCAD MODIFY PASSWORD : NONE

NAME	TYPE	LENGTH	KEY
ENTID	INT	1	YES
SGVEC	IVEC	1	
POSITION	DVEC	3	
TANGENT	DVEC	3	
COMPID	INT	1	YES

CURRENT NUMBER OF ROWS = 64

RELATION : ELEMENT

ELEMID	ELEMTYPE	ELEMNAME	DIM	COMPVEC
10	CELL	MAIN	1	562
30	CELL	CELLA	8	50 114 178 242 306 370 434 498

RELATION : COMPNENT

COMPID	COMPNAME	COMPTYPE	CGVEC	DIM	ENTVEC	ELEMID
562	CELLA	CELL INSTANCE	0 1 0 0 0 1 0 0	1	566	10
50	BICUBIC	SPLINE SURFACE	0 8 0 0 0 1 0 0	8	55 62 69 76 83 90 97 104	30
.	.	.	.	.	. . . .	.
.	.	.	.	.	. . . .	.
.	.	.	.	.	. . . .	.
242	BICUBIC	SPLINE SURFACE	0 8 0 0 0 1 0 0	8	247 254 261 268 275 282 289 296	30
.	.	.	.	.	. . . .	.
.	.	.	.	.	. . . .	.
.	.	.	.	.	. . . .	.

RELATION : CELL

ENTID	CELLXYZ	A_AXIS	B_AXIS	CELLGVEC	COMPID
566	0. 0. 0.	1. 0. 0.	0. 1. 0.	0 0	562

RELATION : ENTITIES

ENTID	ENTTYPE
566	CELL
55	SPLINE
62	SPLINE
69	SPLINE
76	SPLINE
83	SPLINE
90	SPLINE
97	SPLINE
104	SPLINE
.	.
.	.
.	.
254	SPLINE
.	.
.	.
.	.
289	SPLINE
.	.
.	.
.	.

RELATION : SPLINE

ENTID	SGVEC	POSITION			TANGENT			COMPID
55	0	.707	.707	0.	0.	0.	0.	50
62	0	.707	.707	10.	0.	0.	0.	50
69	0	1.	0.	0.	0.	0.	0.	50
76	0	1.	0.	10.	0.	0.	0.	50
83	0	.707	.707	0.	1.	-1.	0.	50
90	0	1.	0.	0.	0.	-1.	0.	50
97	0	.707	.707	10.	1.	-1.	0.	50
104	0	1.	0.	10.	0.	-1.	0.	50
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
254	0	1.	0.	10.	0.	0.	0.	242
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
289	0	1.	0.	10.	0.	-1.	0.	242
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.