Contents

The Development of a Geometric Modeling/Database Management Interface

William J. Rasdorf* Sivand Lakmazaheri† Osama Abudayyeh†

Abstract

In order to transfer data between two software systems used as parts of an overall CAD/CAM system for the design and manufacture of custom orthopedic footwear, we developed an interface program to map the system's inputs and outputs. This paper discusses the NASCAD-RIM Interface version 1.0 for these two systems, the NASCAD geometric modeler and the RIM relational database management system. This interface program for these widely used systems could prove useful to software developers and users.

1 Introduction and Background

The NASCAD-RIM INTERFACE system was devised as a part of a project to develop an integrated CAD-CAM system for the design and manufacture of custom orthopedic footwear[1]. This system utilized a geometric modeler whose main module was the NASA Computer-Aided Design System (NASCAD) and a database management system, the Relational Information Manager (RIM), which was developed by Boeing Commercial Aircraft Company for NASA's Langley Research Center. Using RIM in the footwear CAD/CAM project has helped transfer this NASA-based technology to industry and made available to the project the large body of NASA's RIM expertise.

*Associate Professor, Departments of Civil Engineering and Computer Science, North Carolina State University, Raleigh, NC 27695

†Research Assistant, Department of Civil Engineering, North Carolina State University, Raleigh, NC 27695

[1]This work was sponsored by the National Aeronautics and Space Administration under Grant NAG-1-883 entitled "Engineering Specifications and System Design for CAD/CAM of Custom Shoes–Data Representation and System Integration."

The architecture of the overall integrated CAD/CAM system was presented at the 1988 ASME Computers in Engineering Conference [1]. This paper focuses on the mechanism used to transfer data to and from NASCAD and RIM. The interface system that transfers this data allows other application programs to use RIM to share geometric data with NASCAD. Furthermore, this interface system makes it possible to use a database management system to modify, in a limited way, graphical models created by NASCAD. Presented below is further background information on NASCAD and RIM which will be helpful in understanding this interface system. (For additional details on either NASCAD or RIM, refer to their respective users manuals [2,3].)

1.1 NASCAD

NASCAD, which is the geometric modeler of the software package NEXUS (NASA Engineering eXtendible Unified Software), 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) is capable of procedural definitions, conditional executions, arithmetic and character operations, database manipulation, and graphics operations. These capabilities can be used to develop macros that define and manipulate geometric models.

The structure of the NASCAD database is a tri-level hierarchy consisting of elements, components, and entities [2]. Each element is composed of one or more components, and each component contains two or more entities. As Figure 1 shows, the structure of this database is a double ring, with the nodes of the primary ring consisting of elements and the nodes of the secondary ring consisting of components. Each component's constituent entities are also listed. Figure 1 indicates that NASCAD currently supports four types of elements, eleven types of components, and five types of entities. The NASCAD database can be stored in a file, referred to as a NASCAD data file.

1.2 RIM

RIM has been used as the central database management system of many computer-based analysis systems at the NASA Langley Research Center and elsewhere. It is suitable for most scientific applications since it supports scalar, vector, and matrix attributes whose types include 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 [3].

RIM has two roles in an overall integrated system: first, it transfers data, and second, it stores data. It can help transfer data between a set of heterogeneous application programs and enable each application program to access the data independent of its physical organization in the database. In storing data, RIM can capture information obtained at different stages of design from each application program, thus more easily maintaining the data integrity of the overall design process [4].

2 NASCAD-RIM Interface System Overview

The NASCAD-RIM interface system operates as any interface system in that, for data to be transferred between independent application programs, a data path must exist between two programs. This interface program was developed so that both programs could transmit output data that could subsequently be converted to input data to the other program. Our task was to define a mapping function between NASCAD and RIM so that the NASCAD graphics data could be completely mapped to its corresponding representation in RIM, and vice versa. Figure 2 illustrates how the NASCAD-TO-RIM and RIM-TO-NASCAD interfaces operate.

Below is a description of the interface program in terms of its subsystems and data, and the following section describes its structure.

2.1 NASCAD-TO-RIM Subsystem

This subsystem accomplishes the following:

- 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 data in the corresponding relations in the RIM database.

This subsystem is organized into two sets of modules: one set, FROM-NASCAD, retrieves data from the NASCAD database, and the other set, TO-RIM, stores the data in the RIM database. The NASCAD data structure allows the data extraction process to be modularized 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**, which extract the data associated with NASCAD's elements, components, and entities, respectively. Module C contains a set of submodules (**C1** through **C5**) which extract data associated with the following NASCAD entities: **point**, **arc**, **text**, **cell-instance**, and **spline**, respectively.

The TO-RIM module is composed of eight submodules, named **S**,**T**,**U**,**V**,**W**,**X**,**Y**, and**Z**. These submodules store the data extracted from NASCAD in the following database relations: **elements**, **components**, **entities**, **point**, **arc**, **text**, **cell-instance**, and **spline**, respectively. Figure 3 illustrates the NASCAD-TO-RIM subsystem.

2.2 RIM-TO-NASCAD Subsystem

This subsystem is similar to the NASCAD-TO-RIM subsystem in the types of tasks it accomplishes and in its logical organization. It performs the following tasks:

- simulates the structure of the NASCAD global database (a circular linked list),
- extracts the data from the RIM database,
- converts the data into a form recognizable by NASCAD, and

- stores the data in the NASCAD simulated database.

When all the RIM data is transferred to the NASCAD simulated database, the simulated database is stored in a 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 TO-NASCAD modules stores the extracted data in the NASCAD database.

2.3 Data

This section describes the relational database created in RIM to support the NASCAD graphics data [1]. The RIM database consists of eight low-level relations which correspond to some extent with NASCAD's elements, components, and entities. These relations and the information they contain are listed below, with key attributes being underlined.

- ELEMENT(ElementId,ElementType,ElementName,ComponentVector)
  This relation contains name, type, and component IDs of the NASCAD elements.
- COMPONENT(ComponentId,ComponentName,ComponentType,
  ComponentGraphicsVector,EntityVector)
  This relation contains name, type, graphics information, and entity IDs of the NASCAD components.
- ENTITIES(EntityId,EntityType)
  This relation contains type information for the NASCAD entities.
- POINT(EntityId,PointXYZ,PointGraphicsVector,ComponentId)
  This relation contains information about the POINT entities generated in NASCAD.
- ARC(EntityId,ArcType,StartPoint,CenterPoint,A-axis,B-axis,ArcGraphicsVector,
  ComponentId)
  This relation contains information about the ARC entities generated in NASCAD.
- TEXT(EntityId,Origin,Aspect,Slant,Path,Offset,TextGraphicsVector,TextString,
  ComponentId)
  This relation contains information about the TEXT entities generated in NASCAD.
- CELL(EntityId,CellOrigin,A-axis,B-axis,ComponentId)
  This relation contains information about the CELL INSTANCE entities generated in NASCAD.
- SPLINE(EntityId,SplineGraphicsVector,PositionVector,TangentVector,ComponentId)
  This relation contains information about the SPLINE entities generated in NASCAD.

3 NASCAD-RIM Interface Program Structure

The NASCAD-RIM Interface system is composed of two separate programs: NASCAD-TO-RIM and RIM-TO-NASCAD [5]. The NASCAD-TO-RIM program transforms a **graphical** model created in NASCAD into a **relational** model and stores it in RIM. The RIM-TO-NASCAD program, on the other hand, transfers the data model from RIM back to NASCAD. Both of these programs utilize some of the existing NASCAD and RIM routines. These programs are described below and are illustrated in Figures 4 and 5, respectively.

3.1 NASCAD-TO-RIM Program

This program consists of eight main steps.

1. The NASCAD-TO-RIM interface simulates the internal NASCAD database by invoking the AASYSBGN routine.
2. The SETISUB, INFILENAME, and FILEIO routines are then invoked. (These set the I/O channel to the input mode, define the NASCAD filename, and load the data into the simulated NASCAD database, respectively.)
3. The OPEN$RIM module opens the RIM database and establishes pointers to the appropriate RIM relations via the RMOPEN and RMFIND routines.
4. The TRAVELEM routine is invoked for traversing the circular linked list that represents the NASCAD database structure. (Each node of this linked list is a NASCAD element with its own structure–a circular linked list. A node of this linked list, however, represents a NASCAD component.)
5. The TRAVCMP routine is invoked to process all the NASCAD components by traversing this component-linked list structure.
6. The TRAVENT routine is invoked to retrieve the entities which comprise each NASCAD component. This routine determines the type of each entity and in turn invokes the appropriate module for storing the entity's data in the RIM database. There are five such modules–the POINT$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.
7. The PUTRIM_ENT, PUTRIM_COMP, and PUT$RIM_ELEMENT routines, respectively, are invoked to place the appropriate information in the RIM relations

ENTITIES, COMPNENT, and ELEMENT. (These routines use RMFIND and RMLOAD routines to establish a pointer to the appropriate RIM relation and to load the relation with data, respectively.)

8. The RMCLOS routine is invoked to close the RIM database.

3.2 RIM-TO-NASCAD Program

This program consists of ten main steps.

1. The RIM-TO-NASCAD 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.)

2. The RIM database is opened by calling OPEN$RIM, thus making it possible to transfer data from RIM to NASCAD.

3. 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.

4. The DEFCELLN routine constructs the cell data structure and the routine.

5. LEVEL2 transfers the data from the RIM database to the NASCAD database. (The data is transferred by retrieving all the components of the current graphics cell from the RIM database and storing them in the NASCAD database.)

6. 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.

7. The COMPNENT and LEVEL3 routines extract the entity data from the RIM database and store them in the NASCSAD database.

8. For each entity, LEVEL3 identifies the RIM relation that holds the entity data by calling the ENTITIES subroutine.

9. Then, one of the POINT, ARC, STRING ,CELL, or SPLINE routines is invoked to extract the data from the POINT, ARC, TEXT, CELL, or SPLINE relations, respectively, and to store them in the NASCAD database. (The actual data extraction is done via the RMFIND, RMWHER, and RMGET routines.)

10. 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 the external file. (These last three tasks are done by calling the SETISUBI, OUT FILE, and FILEIO routines, respectively.)

4 Example of the Use of the Interface Program

This section provides a simple example to demonstrate how the interface program works. The example creates a model using NASCAD, transfers the model to RIM by running the NASCAD-TO-RIM portion of the interface, modifies the model using the RIM query language, transfers the modified model back to NASCAD by running the RIM-TO-NASCAD portion of the interface, and displays 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 [5].)

Figure 6 displays the model as it appears in NASCAD before transferring it to RIM. After creating the model it was transferred to RIM by issuing the command:

$ *RUN TORIM*

The program responded by asking the user to enter the name of the NASCAD file to be transferred and the name of the RIM database into which it was to be transferred. From this database the SPLINE relation is shown in Table 1 to illustrate the use of RIM commands.

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 in Table 2. 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 [5].)

5 Summary

This paper briefly described NASCAD and RIM and focused on the interface system that was developed to transfer data between the NASCAD geometric modeler and the RIM relational database management system. This interface system was devised so that other application programs can share data with NASCAD via the database.

Because both NASCAD and RIM are widely used programs that are available from NASA's Computer Software Management and Information Center [6], the software developed at NCSU to interface these programs may be of value and interest to software developers and users. The software is written in standard FORTRAN 77 and is, therefore, highly portable. (Further information about the system may be obtained from the authors.)

6 References

[1] 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, ASME, San Francisco, CA, August 1988. Also in Managing Engineering Data: Emerging Issues, R. Fulton, Editor, ASME, San Francisco, CA, Pages 35-42, 1988.

[2] *NEXUS/NASCAD System Handout*, Cosmic Software Services, October 1986.

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

[4] Rasdorf, W. J., "Extending DBMSs for Engineering Applications," Computers in Mechanical Engineering (CIME), Volume 5, Number 5, Pages 62-69, March 1987.

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

[6] *Software Catalogue*, Computer Software Management and Information Center (COSMIC), Athens, GA, 1986.

ENTID	SGVEC	POSITION			TANGENT			COMPID
55	0	0.707	0.707	0.0	0.0	0.0	0.0	50
62	0	0.707	0.707	10.0	0.0	0.0	0.0	50
69	0	1.000	0.000	0.0	0.0	0.0	0.0	50
76	0	1.000	0.000	10.0	0.0	0.0	0.0	50
83	0	0.707	0.707	0.0	1.0	-1.0	0.0	50
90	0	1.000	0.000	0.0	0.0	-1.0	0.0	50
97	0	0.707	0.707	10.0	1.0	-1.0	0.0	50
104	0	1.000	0.000	10.0	0.0	-1.0	0.0	50
.	.		.			.		.
.	.		.			.		.
.	.		.			.		.
254	0	1.000	0.000	10.0	0.0	0.0	0.0	242
.	.		.			.		.
.	.		.			.		.
.	.		.			.		.
289	0	1.000	0.000	10.0	0.0	-1.0	0.0	242
.	.		.			.		.
.	.		.			.		.
.	.		.			.		.

Table 1: Initial RIM SPLINE Relation

ENTID	SGVEC	POSITION			TANGENT			COMPID
55	0	0.707	0.707	0.0	0.0	0.0	0.0	50
62	0	0.707	0.707	10.0	0.0	0.0	0.0	50
69	0	1.000	0.000	0.0	0.0	0.0	0.0	50
76	0	0.500	0.000	10.0	0.0	0.0	0.0	50
83	0	0.707	0.707	0.0	1.0	-1.0	0.0	50
90	0	1.000	0.000	0.0	0.0	-1.0	0.0	50
97	0	0.707	0.707	10.0	1.0	-1.0	0.0	50
104	0	0.500	0.000	10.0	0.0	-1.0	0.0	50
.	.		.			.		.
.	.		.			.		.
.	.		.			.		.
254	0	0.500	0.000	10.0	0.0	0.0	0.0	242
.	.		.			.		.
.	.		.			.		.
.	.		.			.		.
289	0	0.500	0.000	10.0	0.0	-1.0	0.0	242
.	.		.			.		.
.	.		.			.		.
.	.		.			.		.

Table 2: Modified RIM SPLINE Relation

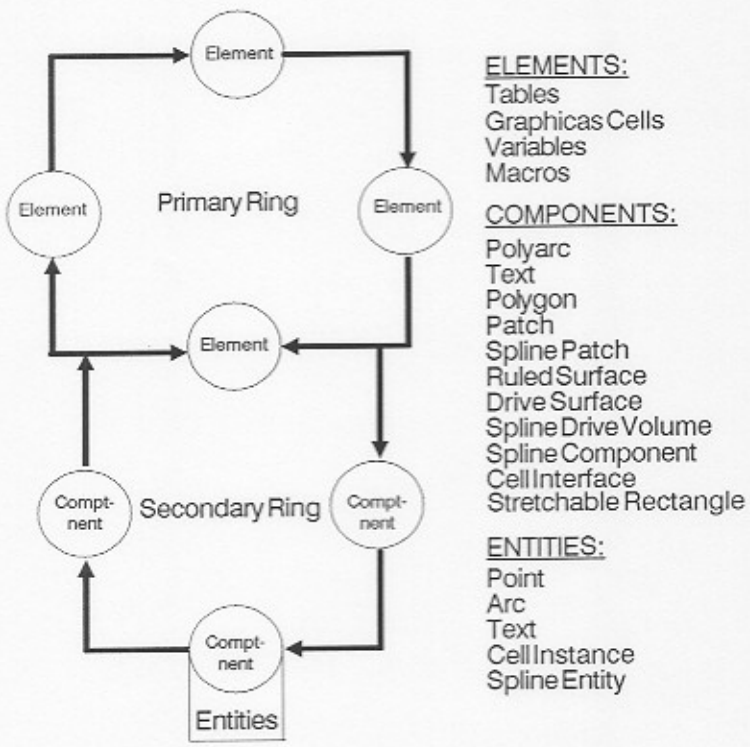

Figure 1: NASCAD Data Structure

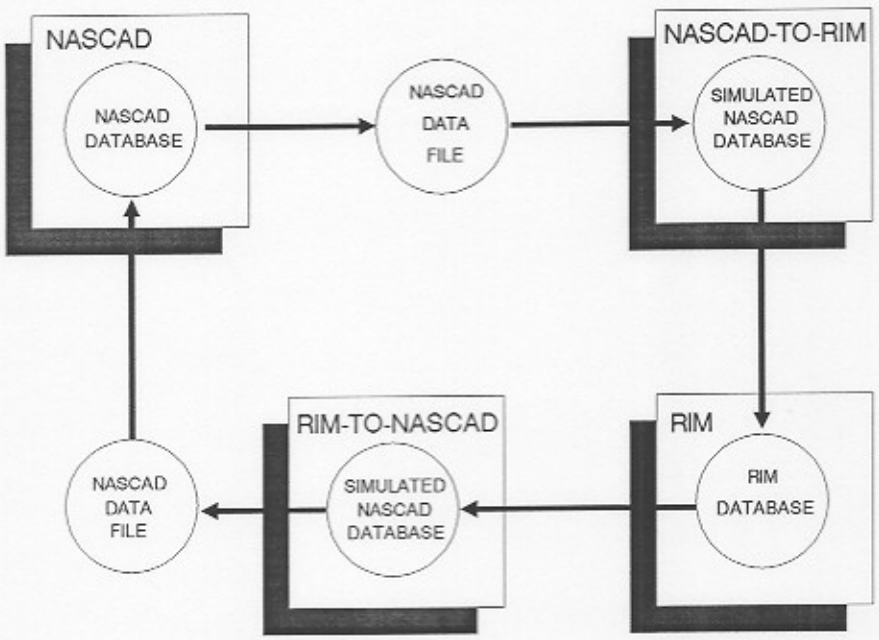

Figure 2: NASCAD-RIM Interface System

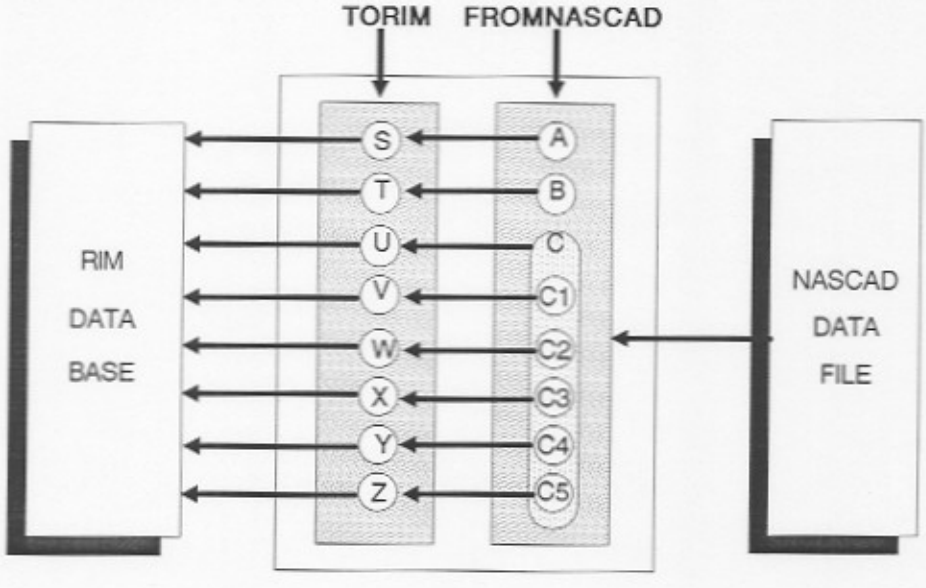

Figure 3: NASCAD-TO-RIM Subsystem

```
NASCADTORIM
    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 NASCADTORIM Program

```
RIMTONASCAD
    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 RIMTONASCAD Program

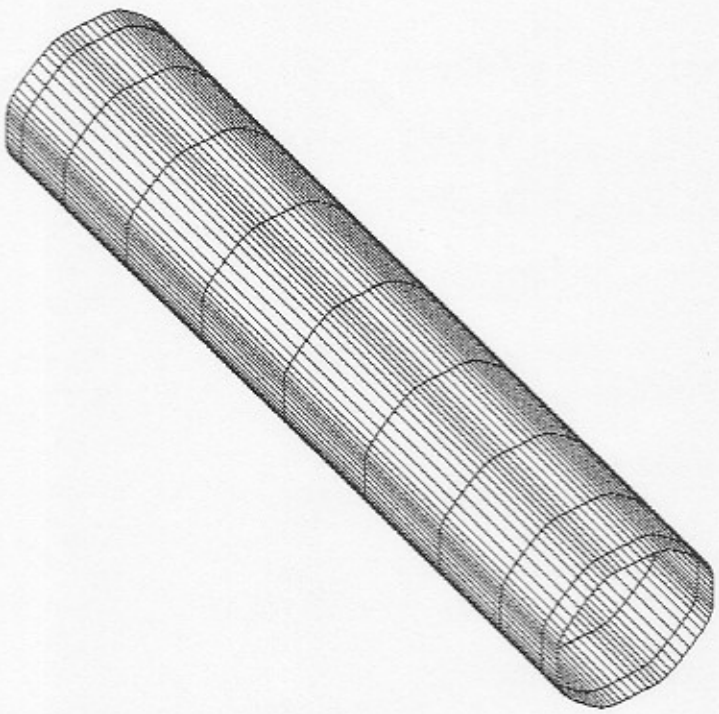

Figure 6: The Model Created by NASCAD

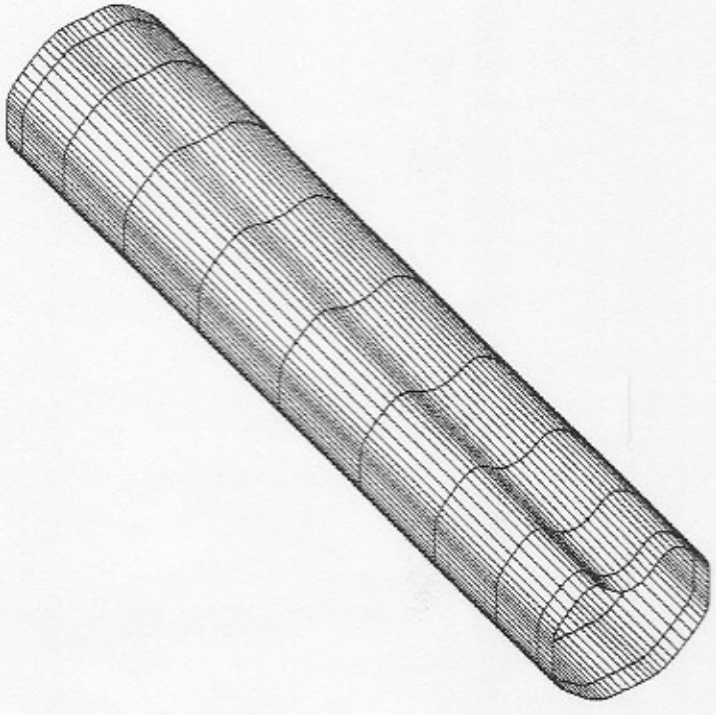

Figure 7: The Model After Modifying It in RIM
and Transferring It Back to NASCAD