7. DESIGN AND DEVELOPMENT OF DATABASE

7.1 Introduction

Computer Aided Process Planning (CAPP) has been recognized as playing a key role in Computer Integrated Manufacturing (CIM). A few CAPP systems have realized or are undertaking interfaces to CAD, CAM and some other computerized systems. The current approaches are mainly trying to interface various separated activities at the design, manufacturing and planning phase. Each of the phases has its own stand-alone relational database and corresponding Database Management Systems (DBMS). There are great difficulties to interface all of these separated activities because of technical problems of software and hardware. In order to realize the final integrated production, it is probably an ideal approach to integrate all of the information involved in producing a product into a database instead of interfacing all of the stand-alone databases. The integrated database may include all the data from design, analysis, drafting, process planning, manufacturing, and NC tool paths as well as project planning, bill of materials and production scheduling information, post-process planning information etc., thus serving as an effective integration between CAD and CAM.

The percentage breakdown of process planner's time spent in performing activities in traditional process planning is as give below:

- Technical decision making 15%
- Data lookup and calculation 40%
- Text and Document preparation 45%

Hence it can be seen that by developing a database which contains information that are required for carrying out the process planning, the time spent by the process planner in data lookup and text and document preparation can be greatly reduced.

7.2 Main Content of the Problem

The efficiency of process planning system depends very much on the quality and extent of data available to it. It requires large amount of data related to manufacturing like tools, machines, cutting parameters, fixtures, raw material, process capabilities, part data, etc. These large data has to be properly organized for easy and quick access as well as maintaining data integrity. The high speed accessibility and cross
checking of the data require a rational selection of the database management system and also proper design of the entire system to minimize the data redundancy.

### 7.3 Objective

The objective of the work is analyze the issues involved in the design and development a centralized database containing information regarding part and manufacturing resources for a Computer Aided Process Planning System. The entire information required to carry out the process planning activity is to be provided by this centralized database.

To accomplish the above objective, the study was focussed on:

- Design aspects of centralized database
- Design of data access procedures – Key issues
  - Resource selection
- Design of User interface.
  - Level of interaction
  - User interface
  - Reports generation

### 7.4 Design issues

The entire information that is required to carry out the process planning activity effectively is to be provided by the database. It has to contain data on cutting tools, raw material, processes, cutting parameters, machine tools, etc. those are pertaining to work environment. In this work study has been restricted to DRDL. This is one aspect that has to be borne in mind in using CAPP software. Extensive customization may be required in using a general purpose CAPP software. The part data that is extracted from solid models by the feature recognition module are stored in the respective database, which is further used by process generation module. This large data has to be properly organized for easy and quick access as well as maintaining data integrity, which requires the proper design of the entire system minimizing data redundancy.

Data entry/retrieval can be enhanced by the design of User Interface forms and also the required reports can be generated. The database is interfaced with other modules by means of Pro *C interface.
A database system is partitioned into modules that deal with each of the responsibilities of the overall system. In most cases, the computer's operating system provides only the most basic services and the database system must build on that base. Thus, the design of the database system must include consideration of the interface between the database system and the operating system.

The functional components of the database system are shown in Fig. 7.1.

**7.5 Overall System Structure**

A database system is partitioned into modules that deal with each of the responsibilities of the overall system. In most cases, the computer's operating system provides only the most basic services and the database system must build on that base. Thus, the design of the database system must include consideration of the interface between the database system and the operating system.

The functional components of the database system are shown in Fig. 7.1.
7.6 Database Design

On the basis of the requirement analysis a logical structure [4] is adopted in this work.

7.6.1 Study of existing system

Before designing the database for any application, the existing system has to be studied and the information has to be recorded properly. This is usually done by means of Data Flow Diagrams (DFD). It is a pictorial way of showing the flow of data into, around and out of a system. The existing system of process planning practiced at DRDL has been studied and the information flow in the system is shown in Fig.7.2.

Firstly, the job specifications are read from the part drawing. From the part and process capability data, the operation to be performed to finish each feature in the part is decided.

Once operation is decided, then for each feature of the part best possible cutting tool is selected for carrying out the operation.

Then based on the workpiece material, cutting tool material and type of operation, the process parameters are selected.

For the cutting tool selected and the process parameters selected, a suitable machine tool has to be selected that can accommodate the cutting tool and workpiece and that can provide the speeds and feeds which are previously selected.

These processes are sequenced using some sequencing algorithms and finally generate the required process plan.

The items represented in ellipses are called external entities, which represent sources of data that enter the system or the recipients of data that leave the system.

The items represented rectangles numbered 1,2,3,4,5, and 6 are called data stores.

The data items represented by larger rectangles are called processes in which data is manipulated by being stored or retrieved or transformed.

The arrow marks represents the data flows, which shows the movement of data between other components of the diagram.
7.6.2 Phase of Database Design

7.6.2.1 Requirements collection and analysis

A set of user requirements has to be considered in this phase. These include data requirements as well as functional requirements of that application. These consist of transactions that will be applied to database and they include both retrievals and updates.

7.6.2.2 Conceptual Schema

The conceptual schema is a concise description of the data requirements and includes detailed description of data types, relationships and constraints. The data flow diagram (DFD) for process planning is shown in Fig.7.2, which shows how exactly the information flow takes place in deriving at a process plan. After the conceptual schema has been designed, the basic data model operations can be used to specify high-level transactions corresponding to the DFD identified during functional analysis.

Fig.7.2 Data Flow Diagram for the existing system
7.6.2.3 Logical Model

The data model describes data as entities, relationships, and their attributes. In the CAPP domain the various entities identified are machine tools \{MT\} cutting tools \{T\}, operations \{P\}, part and features \{PF\}. Also there exists other entities but these are used to implement the functions provided by certain other entities and are not significant enough to be included in the ER model. The Entity Relationship diagram (ER diagram) drawn for the CAPP system is shown in Fig.7.3. The rectangle in the diagram represents an entity and diamond box represents the relations between entities or interaction between the entities like,

Many of the data entities used in process planning have many-to-many (M: N) relations like

- A machine can perform more than one type of operation
- A tool can perform more than one type of operation
- An operation could be performed by more than one machine
- An operation could be performed by more than one tool.
7.6.2.4 Derivation of entities in each subsystem

After the mapping of relational data model as shown in Fig. 7.3, this stage involves establishing the functional dependencies between different attributes of an entity and derivations of relations for each subsystems by applying the normalization concepts.

Normalization of data can be looked on as a process during which unsatisfactory relation schemas are decomposed by breaking up their attributes into smaller relation schemas that possess desirable properties. One objective of the normalization process is to ensure that the update anomalies like inserting, modifying, deleting do not occur.

General steps for carrying out a normalization process in the selection of a drill are as follows:

The set of attributes that are participating in the schema 'R' is:
\[
R = \{\text{Drilltype\_code, Drill\_class, dia\_min, dia\_max, Drilltype, dep\_min, dep\_max, Cutting\_fluid, hole\_tol, Roughness, Shanktype, Material\_type, Rank, Drill\_code, Drill\_dia, Shank\_dia, total\_tool\_length (11), ISO\_length (13), recommended\_drill\_length (14), carbide\_grade}\}
\]
dia_min, dia_max, Drill_type, depth_min, depth_max, cutting_fluid, hole_roughness, shank_type are functionally dependent on the combination of Drilltype_code, Drill_class

Material_type is multivalued dependent on Drilltype_code, Drill_class and rank is functionally dependent on material_type.

Drill_code is multivalued dependent on the combination of Drilltype_code and Drill_class.

Drill_dia, shank_dia, 11, 13, 14, carbide_grade are functionally dependent on the Drill_code.

The functional dependencies on the above relation schema are,

\[ F = \{\text{Drilltype_code, Drill_class} \rightarrow (\text{Dia_min, Dia_max, Drilltype, Dep_min, Dep_max, Cutting fluid, hole_tol, Roughness, Shanktype, Material type}), \]
\[ \rightarrow \text{Material type (Rank)} \rightarrow \text{Drill_code (Drill_dia, shank_dia, l4, carbide_grade)}\]  

In the above relation (Drilltype_code, Drill_class) is considered as primary key.

Based on these functional dependencies, the unsatisfactory relations are divided into satisfied relations as shown in Fig.7.4
Normalization of the relation yields three relations that are in 4NF because every determinant is a candidate key and there exists no nontrivial MVD.

The design approach used here is to divide the total system into subsystems and then integrate individual view models into a database model. The main reason for such division and recombination is to avoid handling a universal relation containing a very large number of attributes.

The last phase of database design involves the transformation of a database model into a concept schema, which can be defined by the data definition language (DDL) of a specific DBMS. Once the database model is described with a DDL schema, loading the database file creates the actual physical database.

7.7 Database Development

The logical structure of the database derived from the design phase is implemented in ORACLE 7.0 by creating tables, establishing primary keys and maintaining referential integrity constraints. The DDL statements used to implement a drill tool database is shown below:
CREATE TABLE DRILLTYPE_SE (DRILL_TYPE VARCHAR2 (20),
DRILL_CLASS CHAR (1), DIA_MIN NUMBER (4,2), DIA_MAX NUMBER (4,2),
DEPTH_MIN NUMBER (5,2), DEPTH_MAX NUMBER (5,2),
DRILLTYPE_CODE VARCHAR2 (6), CUTTING_FLUID VARCHAR2 (10),
PRIMARY KEY (DRILLTYPE_CODE, DRILL_CLASS));

CREATE TABLE DRILLTYPE_MAT (DRILLTYPE_CODE CHAR (6),
DRILL_CLASS CHAR (1), MATERIAL_TYPE CHAR (20), RANK NUMBER (1),
PRIMARY KEY (DRILLTYPE_CODE, DRILL_CLASS, MATERIAL_TYPE)
FOREIGN KEY (DRILLTYPE_CODE, DRILL_CLASS));

CREATE TABLE DRILL_SE (DRILLTYPE_CODE CHAR (6), DRILL_CLASS CHAR (1),
DRILL_CODE CHAR (25), DRILL_DIA NUMBER (4,2), SHANK_DIA NUMBER (4,2),
L1 NUMBER (5,2), L3 NUMBER (5,2), L4 NUMBER (5,2),
CARBIDE_GRADE CHAR (3), PRIMARY KEY (DRILLTYPE_CODE, DRILL_CLASS, DRILL_CODE));

By loading these command files and running in ORACLE environment the drill tool database can be created. By following the similar procedure all the database relations can be created after obtaining the parameters from the conceptual design stage, also the integrity constraints can be maintained by establishing foreign keys.

The following are the databases that are designed and implemented:

- Part database
- Feature database
- Blank database
- Process database
- Tool database
- Cutting parameters database
- Machine tool database
- Fixture database
- Gauge database
7.7.1 Part Database

This database contains information regarding the Part like Part no., Part name, Material Type, Dwg. no, Assembly name etc. The user enters this information on the arrival of the any new work order from the part drawing as shown.

<table>
<thead>
<tr>
<th>Part no</th>
<th>Part name</th>
<th>Material type</th>
<th>Dwg. no.</th>
<th>Assembly name</th>
</tr>
</thead>
</table>

7.7.2 Feature Database

After modeling of the Part in CADDS5, the extracted features will be stored in the database with attributes like feature name, feature identification code, feature dimensions, tolerances on feature dimensions etc.

<table>
<thead>
<tr>
<th>Feature id</th>
<th>Feature name</th>
<th>feature_dim</th>
<th>tolerances on feature_dim</th>
</tr>
</thead>
</table>

7.7.3 Blank Database

This database contains information regarding available blank size, blank shape, blank material, hardness, material condition and other details, which are useful for blank selection.

<table>
<thead>
<tr>
<th>Blank_code</th>
<th>blank_size</th>
<th>blank_shape</th>
<th>blank_matl.</th>
<th>hardness</th>
<th>matl_condition</th>
</tr>
</thead>
</table>

7.7.4 Process Database

It contains information regarding various processes, the tolerance and roughness that can be achieved with each type of process. This will be useful in the process selection for each feature to be processed. Each process is identified by unique code that will help in the identification of the process.

<table>
<thead>
<tr>
<th>Process_code</th>
<th>type of process</th>
<th>tol_min</th>
<th>tol_max</th>
<th>rough_min</th>
<th>rough_max</th>
</tr>
</thead>
</table>

7.7.5 Tool Database

The Tool Database contains information regarding general processing tools like Boring, Drilling, Endmilling, Facemilling, Sidemilling, Parting, Grooving, Turning etc., and also the inserts required for the cutters with their geometry and material. The database provides details like type of tool, ISO description, tool material
characteristics, etc. Data for the above tools from standard manufacturers 'SANDVIK' is made available.

<table>
<thead>
<tr>
<th>Tool_type</th>
<th>ISO description</th>
<th>tool_material</th>
<th>inserts</th>
<th>ins_geom</th>
<th>ins_matl.</th>
</tr>
</thead>
</table>

7.7.6 Cutting Parameters Database

This database contains information regarding cutting speed, cutting feed for various work material and tool material combinations. Separate relations have been created for each category of tools for determining the cutting parameters.

<table>
<thead>
<tr>
<th>CMC_no</th>
<th>speed_min</th>
<th>Speed_max</th>
<th>feed_min</th>
<th>feed_max</th>
</tr>
</thead>
</table>

7.7.7 Machine Tool Database

This database provides information about all the work centers available in the machine shop, which can be useful in selecting the required work centers based on the feature to be processed. It contains information like speed, feed, and power requirements, surface finish that can be attained and the tolerance specifications of each individual machine.

Workcenters that are considered are lathe, boring, drilling, honing, jig boring, surface grinding, milling, cylindrical grinding machines etc.

<table>
<thead>
<tr>
<th>machine_code</th>
<th>manufacturer</th>
<th>specifications</th>
</tr>
</thead>
</table>

7.7.8 Fixture database

The fixturing used is modular fixturing, which is used to reduce the time required for the preparation. The fixtures used are Universal Sectional Fixtures (YCIT). For lathe, milling, drilling, and other machining operations, as well as for inspection operations they are repeatedly and diversely assembled out of ready-made standardized interchangeable and wear resistant part and units. This database contains details of fixture elements like plates, rest and guiding elements, connecting strips, prisms etc.

The data stored for plates is as shown below:

<table>
<thead>
<tr>
<th>Part_no</th>
<th>length</th>
<th>Width</th>
<th>height</th>
<th>no.of slots</th>
<th>type of slot</th>
<th>weight</th>
</tr>
</thead>
</table>
7.7.9 **Gauge database**

This database contains information about various gauges and inspection instruments that are available in the stores that can be used for the inspection of the part produced. The database for thread gauges is only one to be implemented.

7.8 **Resource Selection**

After the design and development of database, the operations to be done on each feature are decided based on process capability database and then resources like blank, cutting tools, cutting parameters, machine tools are selected from the database for each operation. This is usually done by conditional queries submitted in the form of programs, written in PL/SQL. These programs can be used in SQL *Forms as well as can be embedded in 'C' language so that it accepts the inputs from other modules and supplies the required information about the resources existing, which aids in the process generation.

7.8.1 **Selection of tools**

Once the process is decided for a feature, then the best matching tool has to be selected by entering the basic information about the feature. It provides data like tool ISO code, tool length, diameter, inserts, insert material etc. Separate databases are developed for drill tools, bore tools, turning tools, face milling cutters, end milling cutters, side milling cutters, parting and grooving tools of SANDVIK. The methodology involved in the selection of drill tool is shown in Fig.7.5

7.8.2 **Selection of cutting parameters**

The manufacturer of the cutting tools usually supplies the cutting parameter data like speed, feed, depth of cut, etc. These data is stored in cutting parameter database and separate cutting parameter tables are available for each of type of tools. Firstly the cutting tool type is selected, and then Coromant material classification for the workpiece is selected from the CMCno table. Then for the combination of Drilltype code, drill diameter, and CMCno the cutting speed and feed is selected from the cutting parameter database supplied for each type of tool. The cutting parameter selection is done in the same way for other operations like turning, milling, etc. The methodology involved in the selection of drill cutting parameters is shown in Fig.7.6.
Fig. 7.5 Algorithm for Drill selection
7.8.3 Selection of Machine

Once the process, cutting tool and process parameters are selected then the machine tool on which the workpiece can be loaded to carry out the operation can be selected. The process generation module decides upon the class of machine to be used on the process to be carried out. Then based on workpiece dimensions, accuracy needed, and other factors like speed, best matching machine can be selected from the machine database, which provides data like machine code, machine manufacturer and other details. The various machine classes considered are lathes, milling machines, grinding machines, jig boring machines, boring machines, honing, lapping machines etc. The methodology involved in the selection of drill machine is shown in Fig. 7.7.
7.8.4 Selection of Milling Cutters

The typical selection procedure for milling cutters is described below:

i) **SIDE MILL**

ii) **FACEMILL**

iii) **ENDMILL**
i.a) Selection of sidemilling cutter

- \texttt{ssg\_cuttype}
- \texttt{sm\_cut\_opr}
- \texttt{sm\_cutter}
- \texttt{sm\_ins}

\texttt{primary key(sm\_cuttype, sm\_cuttype\_code) *}
\texttt{foreign key(sm\_cuttype, sm\_cuttype\_code) *}
\texttt{primary key(sm\_cuttype, sm\_cuttype\_code, opr\_type)}
\texttt{foreign key(sm\_cuttype, sm\_cuttype\_code) *}
\texttt{primary key(sm\_cuttype, sm\_cuttype\_code, sm\_cutter\_code)}
\texttt{foreign key(sm\_cuttype, sm\_cuttype\_code, sm\_cutter\_code, sm\_ins\_code) **}

\texttt{references from (sm\_ins)}
\texttt{primary key(sm\_cuttype, sm\_cuttype\_code, sm\_ins\_code, cmg\_code)}

A sample inputs and outputs given for the above selection is shown below:

**Inputs:** operation type - cutting off
- \texttt{dia} - 80
- \texttt{depth} - 17
- \texttt{width} - 2.08

**Output:**
- \texttt{Cutter\_type} - T-MAX Q CUTTER
- \texttt{Cutter\_code} - 330.20-080020-220
- \texttt{Ins\_code} - 330.20-20-AA
- \texttt{Cmg\_code} - GC 235

The records outputted may be one or more than one depending on the condition to be satisfied and the available data. The \texttt{Cmg\_code} in the output will tell the material of the tools (according to ISO it is given). This also becomes a deciding factor for the selection of suitable cutter. The selection procedure is shown in Fig. 7.8
i.b) Sidemill cutting parameters

\[
\text{sm\_cut\_con} \quad \text{primary key(sm\_cuttype, sm\_cuttype\_code, cmc\_no, cmg\_code)}
\]

\[
\text{blank\_detail} \quad \text{primary key (blank\_code)}
\]

\[
\text{cmc\_no} \quad \text{primary key (cmc\_no)}
\]

The selection procedure is described in the Fig. 7.9. A sample of data inputted and the output obtained is shown below:

**Input:**
- Blank\_code: A001
- Cutter\_type: T-MAX Q CUTTER
- Cutter\_code: 330.20
- Cmg\_code: GC235

**Output:**
- Speed: 155 m/min
- cmg\_code: GC235
- cmc\_no: 1.1
- Hardness: 175 BHN
- Cut\_type\_code: 330.20
Fig. 7.8 Algorithm for sidemilling cutter selection

START

Get Operation type, diameter of cutter, depth, width

Select SM-CUTTYPE, SM-CUTTYPE_CODE from SM-CUT-OP by matching input operation type and then by establishing a joining condition fetch all the corresponding values from database

FETCH one record into Variables

Match with inputs if data found

YES

Print Cutter-code, Cutter-type, Insert-code, CMC-code

STOP

DATABASE
In the same way the selection is made for the other cutting tools.

### 7.9 Design of User Interface

The key goal of interface design is to maximize the usability of system by ensuring that users can carry out their tasks effectively and efficiently. The application developed offers two types of user interfaces: i) interactive ii) Non interactive. Interactive interfaces include menus and forms. Non interactive interfaces include report generation and the output resulted through the queries (SQL).
**Fig. 7.10 Drill selection form**

<table>
<thead>
<tr>
<th>Drilltype Code</th>
<th>Drill class</th>
<th>Drill code</th>
<th>Drill dia</th>
<th>length</th>
</tr>
</thead>
<tbody>
<tr>
<td>R 410.5</td>
<td>A</td>
<td>R410.5-1100-60-01TIN</td>
<td>11</td>
<td>74</td>
</tr>
</tbody>
</table>

**Fig. 7.11 Drill cutting parameter selection form**

<table>
<thead>
<tr>
<th>Cuttype code</th>
<th>CMC No.</th>
<th>Material Type</th>
<th>Hardness (BHN)</th>
<th>Speed (m/min)</th>
<th>Dia. (mm)</th>
<th>Feed (Mm/rev)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R 410.5</td>
<td>11</td>
<td>ALLUMINIUM</td>
<td>175</td>
<td>40</td>
<td>11</td>
<td>0.77</td>
</tr>
</tbody>
</table>
### 7.9.1 Forms Based User Interface

Forms facilitate users for easy entry and retrieval of data. Combining blocks will generate forms. Each block represents one or more tables. The sample forms for drill selection, drill cutting parameters selection and drill machine selection are shown in Fig.7.10, Fig.7.11, and Fig.7.12 respectively. The upper block represents input block and lower block represents output block. Each block is collection of fields. The input block accepts input from the user/process planner and when trigger fires, the logic procedure written in PL/SQL will be executed and the output will be fetched into output block.

### 7.9.2 Report Generation

Various Reports are generated using SQL *Reporter, to carry out transactions and to provide hard copies of data that is retrieved from the database, based on conditional queries.

The reports generated consists of three basic components

1. Page layout specifications
2. Data extraction from databases using queries
3. Data formatting
Writing queries to fetch data from the database, and controlling the format of the display can generate the required reports. Also runtime variables can be added whose value is given as input at the time of execution of the report. The output of the reports can be directed to the screen or to a printer.

The various reports generated are

- Project work order
- Material demand form
- In office notice
- Job completion report
- Inspection card
- TCR tool card
- Tool requisition form
- Master list
- Process sheet

7.9.2.1 Project work order

This report is generated by the Project Group and sent to the Process-planning department. These data are stored in the work order detail table. The input parameters for this report are workordno and jobno. For the corresponding workordno and jobno the project, project group, work order date, date required, and no. of items in the work order, the item code with the part description, quantity required, drawing no. etc., will be received from the work order details table.

7.9.2.2 Material Demand Form

This Report is generated by the Process Planning Department and sent to the Material Cutting Section. One copy will be returned to the Process Planning Department to give the feedback about the availability of the material. The input parameters for this report are work_ord_no and job_no. And for the corresponding parameters it retrieves the material code with its specifications from the Blank detail table for each item and the quantity required from the Work Order detail table. A sample material demand form is shown in Fig. 7.13
Demand reed, by stores Mat. To be given to cutting section Mat. Actually supplied Rack No. Cutting material to be ready on

Wk ord No: DRDL/PYD/34/78 Project: PRITHVI
Job No: 4387 Date of demand : 23 Jan '99 P Group : PYD

<table>
<thead>
<tr>
<th>Item No.</th>
<th>Blank_code</th>
<th>Item_name</th>
<th>Qty. required</th>
<th>Qty. issued</th>
<th>Sig.</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2</td>
<td>4200930</td>
<td>HF15 75*30LEN</td>
<td>100</td>
<td>80</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CERTIFICATE
Demand Initiated by:
Note: The above mentioned items certified that the items so mentioned in the demand are demanded for bonafide Government work
Fig 7.13 Material Demand Form

7.9.2.3 In office notice
This report is generated by the Process Planning Department and sent to the Project Group based on the feedback from the Material cutting section if the material to process the component is not available. The input parameters for this report are work_ord_no. and job_no. For the corresponding job_no. it compares the quantity required and quantity issued for each item in the work order detail table and if the material is not sufficient it will indicate the quantity deficiency with the material specifications.

7.9.2.4 Job Completion Report
This report is generated by the Process Planning department and sent to the Project Group after completion of the job. The input parameters for this report are work_ord_no.
7.9.2.5 Tool Card
The Process Planning department sends this to the Tool Eng. Department. If it is a special tool and has to be manufactured by the tool engg. dept. this report is generated. Taking input as TCR No this card generates the entire details about the tool like job no., item no. for which it is manufactured, designer, project, project group, work order, tool type, work material, tool material, radius, purpose etc..

7.9.2.6 Master List
The Process Planning department and sent to the Production control department to carry out the scheduling of the job is generating this Report. It gives details like the item no., their nomenclature, quantity required, operation no. for each item, and the machine required to carry out the operation by taking work order no. and job no. as input parameters.

<table>
<thead>
<tr>
<th>Job No.</th>
<th>Item No.</th>
<th>PROCESS SHEET</th>
<th>PEG, DRDL, HYDERABAD</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Drawing No.</th>
<th>Item Name</th>
<th>Description</th>
<th>Assy. Name</th>
<th>Division</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKT 192</td>
<td>EN 24 DIA 20 LEN 120</td>
<td>: LIFTING 111G</td>
<td>:</td>
<td>LPD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Process Planner</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>K N Srinivas</td>
<td>PRITHVI</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Oper. No</th>
<th>Feat. No</th>
<th>Base Feat.</th>
<th>Sub feat.</th>
<th>Face</th>
<th>Oper.</th>
<th>M/c</th>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig.7.14 Process Sheet

7.9.2.7 Process sheet
This report is being generated by the Process Planning department and sent to the shop floor for manufacture of the component. It gives details like Drawing no., Material used, Process planner, Part description, Operation no., Machine code, Operation details, Speed, Feed, Tool no., Depth of cut, set up times and operation
times by taking Job no. and Item no. as input parameters. Thus all the process plans are stored in the database with the job no. and their corresponding item no. Hence the combination of Job no. and Item no. is unique and there cannot be any duplication of this combination. A sample process sheet format that is generated is shown in Fig. 7.14.

### 7.10 Menu Based User Interface

SQL*Menu is a productivity tool that provides a single Menu Interface for running Multiple data-processing tools i.e., display all the choices available to an operator. This will save the time for data look up and calculation and technical decision making. The menus created are shown in Fig. 7.15 and Fig. 7.16.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Retrieve&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single Point&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi Point&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCNR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End Mill</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Face Mill</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Side Mill</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slitter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grooving</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drill</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DRILL**

Application: CAPP  

< REP>

---

Fig. 7.16 User Interface Form 2

### 7.10.1 Interfacing with other modules

The various resource selection procedures written in PL/SQL are supplied to embed in Pro *C program in order to achieve the complete integration of all the modules of CAPP enabling the free flow of data from one module to another. These Pro *C programs are first precompiled using the pre-compiler and it is then compiled using 'C' compiler.

### 7.11 Conclusion

A framework for the database design is presented in this chapter. Application of these ideas to R&D laboratory is also presented. The data retrieval of tools, cutting parameters, machines etc., for various processes to be carried out on different features have been tested. The necessary reports for transactions have been generated.

Apart from using as an integrated module in CAPP, this centralized database can also be used as a stand alone in any manufacturing environment to save time for the data lookup and technical decision making and document preparation. This database can be easily updated to adopt the new changes. Adding new databases to the existing tool, cutting parameters and machine libraries can further strengthen the database.