THE DEVELOPMENT OF A METALLURGICAL CAPP SYSTEM FOR LARGE STEEL FORGINGS

WENZHONG DENG

April 1995

A thesis submitted in partial fulfilment of
the requirements of Sheffield Hallam University for
the Degree of Doctor of Philosophy

Sheffield Hallam University in Collaboration with
the Sheffield Forgemasters Engineering Limited

Sheffield Hallam University

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank my supervisor, **Professor A. W. D. Hills,** for his excellent supervision, help, encouragement and suggestions, which he has given at all the stages. In addition, I would also like to thank my second supervisors, **Dr. G. A. Honeyman** and **Mr. D. E. Eaton** for their help at various stages and particularly for the necessary facilities provided for this work, and the financial support from Sheffield Forgemasters Engineering Ltd. and Sheffield Hallam University is greatly acknowledged.

I am indebted to many colleagues at Forgemasters Engineering Ltd, a subsidiary of Sheffield Forgemasters, whose knowledge has been used to convert the concept described in this thesis into an operational system. And I should also thank **Mr. Peter Birtles**, the managing director, for his support to the project.

I would like to show my appreciation to the many technicians, secretaries, librarians, lecturers and other members of staff and researchers at Sheffield Hallam University for their help, assistance and services during my studies.

Finally, I would like to record my thanks to **Leiya** and **Chao**, my wife and seven-year-old daughter. They stayed out of the many hours of my time involved in producing this thesis; but were always there when I needed them. These two made it all worthwhile. And my sincere appreciation goes to my sister and her family for their considerable help and closest companionship throughout my studies in the UK.

DECLARATION

I declare that the results obtained and the theories developed during the course of this research are to the best of my knowledge original, except where the reference is made to the work of others.

The data and manufacturing knowledge that actually appears in this thesis are similar to the sponsoring establishment's practice, not its actual practice. This device has been adopted in order to maintain commercial security, the sponsoring establishment being a world leader in the manufacture of large steel forgings, whilst still allowing the research issues to be discussed.

ABBREVIATIONS AND KEY WORDS

AI - Artificial Intelligence.

CAPP - Computer Aided Process Planning.

CAD - Computer Aided Design.

CAM - Computer Aided Manufacturing.

CBP - Case-Based Planning

CIM - Computer Integrated Manufacturing.

DBMS - DataBase Management System

ES - Expert Systems.

GT - Group Technology.

IKBS - Intelligent Knowledge Based System.

KBS - Knowledge Based System.

NC - Numerical Control.

PPL - Process Planning Logic.

SQL - Structure Query Language

QA - Quality Assurance.

SME - Society of Manufactory Engineers

KEY WORDS: Computer Aided Process Planning (CAPP), Artificial Intelligence (AI) in engineering, Manufacturing Automation, Computer Integrated Manufacturing (CIM), Hierarchical Planning, and Computer Aided Manufacturing (CAM).

(Abstract)

The Development of a Metallurgical CAPP System for Large Steel Forgings

by Wenzhong Deng

The development of CAPP (Computer Aided Process Planning) systems promises improvement to the design efficiency and quality of process plans, whilst maintaining knowledge for future developments. Although considerable progress has been made in Computer Aided Process Planning, most of the systems developed or under development so far are limited to one manufacturing operation and to planning for an individual part design. The systems lack an overall structure for real manufacturing practice. This thesis examines the problems involved in the planning of an extensive manufacturing task involving many different processes including chemical and thermal treatments. On the basis of the evaluation of the manufacturing system in the collaborating company, an intelligent database system has been designed to solve metallurgical process planning problems involved in the manufacture of large steel forgings. In this CAPP database system, two hierarchy control levels involving a number of local planning areas have been adopted to allow the development of process sub-plans as well as supporting engineering data. All the process sub-plans have been integrated into a single system rather than isolated as separate entities within the overall metallurgical process planning system together with quality assurance control and other functions. These sub-plans, however, are planned and modified in the separate planning areas, the development being conducted on facsimile data records. Only when each sub-plan has reached a satisfactory state of development is it issued - made available to the overall system - by transferring the facsimile records into the system data files, the facsimile records then being discarded.

Metallurgical process knowledge and rules have been incorporated into the database. These allow the system to assist users to make decisions and achieve final desired process plans. A versional approach has been developed to organise and control the stage by stage evolution of issued process plans within this complex steel forging environment. The use of separate planning areas and local facsimile records allows the modification of sub-plans already issued to be undertaken on a step by step but secure basis. A fully operating authorisation system controlling access to the data and the deletion or modification of records has been achieved. This is essential in a CAPP

system of this type in which historical decisions, or approved rules based on historical experience, are presented to the users as the basis to make new decisions.

The work has been extended to explore external enhancement of the central database system with an expert system and with specially written C ++ programmes. The system architecture needed to support this link is described, and issues raised by the enhancement that relate to the overall control are then addressed.

The final part of the thesis examines the limitations of the method that has been developed and discusses difficulties involved in implementing a CAPP system in a large concern involved in the 'bespoke' manufacture of complex engineering artifacts on a one-off design basis.

CONTENT

		PAGE
TITLE		i
ACKNOWLEDGEMENT		ii
DECLARATION		iii
ABBREVIATION AND KEY WORDS		iv
ABSTRACT		v-vi
LIST OF FIGURES		x-xi
INDEX OF APPENDICES		xii-xv
CHAPTER 1	**INTRODUCTION**	1
CHAPTER 2	**LITERATURE SURVEY**	5
2.1	Process Planning in Manufacturing	5
2.2	CAPP Systems Approaches	7
2.2.1	Variant Approach	7
2.2.2	Generative Approach	8
2.2.3	The Semi-generative Approach	10
2.3	Techniques in CAPP Research	10
2.3.1	AI Techniques	10
2.3.2	Knowledge Capturing & Representation	12
2.3.3	Case-Based Planning	14
2.3.4	Coupling of a Database with an Expert System	16
2.4	Process Planning for Quality	19
2.5	Summary	20
CHAPTER 3	**PROJECT BACKGROUND AND TECHNIQUES**	23
3.1	Manufacturing Domain	23
3.2	Two-level Hierarchical Control	25
3.3	Relational Database System	29
3.4	The AI-Database Approach	30
3.5	Strategies in the Development of the System	32
CHAPTER 4	**COMPLETED WORK ON FEL-CAPP SYSTEM**	34
4.1	Introduction	34

4.2	Manufacturing Function Integration	35
4.2.1	Simplified Logical Structure	36
4.2.2	Versional Data File Requirements	36
4.3	Metallurgical Functions	40
4.3.1	Physical Implementation of Data Structure	40
4.3.1.1	Links between Planning Areas through the Family of Status Files	44
4.3.1.2	Functions of the Forging Status File	48
4.3.2	Facsimile File Approach	49
4.3.3	Process Planning and Data Manipulation	51
4.3.3.1	Forging Definition Planning Area	52
4.3.3.2	Manufacturing Plan Planning Area	58
4.3.3.3	Quality Plan Planning Area	64
4.3.3.4	Steel Order Planning Area	68
4.4	Quality Assurance Functions	70
4.4.1	Integration of Quality Assurance Functions	71
4.4.1.1	Specification Control	71
4.4.1.2	Process Plan Distribution Status	74
4.4.2.	Automation of QA Functions	77
4.4.2.1	Internal Document Distribution Check	77
4.4.2.2	Customer Document Status	78
4.5	Test Results Functions	79
4.6	Intelligent Database System Approach	82
4.6.1	Specification Manipulation	82
4.6.2	Steel Melting Range Design	84
4.6.3	Approval of Steelmaking Procedure & Cast Analysis	91
4.6.3.1	Approval of Cast Analysis	91
4.6.3.2	Approval of Steelmaking Procedures	92
4.7	Summary	96
CHAPTER 5	**DATABASE ENHANCEMENT**	98
5.1	Introduction	98
5.2	Coupling of the Database with an ES	98
5.2.1	System Architecture	98
5.2.2	Operation of the Coupling	104
5.3	Database Enhanced with External Program	105
5.3.1	Automation of Hydrogen Degassing Process Design	105
5.3.2	Theoretical Base for Hydrogen Degassing Calculations	106

5.3.3	Architecture of the Experimental System	119
5.3.4	Operation of the System Enhanced with Programme	122
5.4	Summary	123
CHAPTER 6	**OPERATING EXPERIENCE**	124
6.1	The FEL CAPP System	124
6.2	The Enhanced Database System	127
CHAPTER 7	**DISCUSSION**	131
7.1	Introduction	131
7.2	Versional Approach	131
7.3	Authorisation Control	134
7.4	Incorporating Metallurgical Knowledge	135
7.5	Variant CAPP System Approach	137
7.6	Knowledge Development Process	138
7.7	Hierarchical Process Planning	140
7.8	Comparison of Rule Templates vs. Expert System	141
7.9	Evaluation of the Use of an External Programme	143
7.10	Assessment of the FEL CAPP System	144
CHAPTER 8	**CONCLUSIONS AND FUTURE RESEARCH**	149
REFERENCES		153-162

LIST OF FIGURES

Figure 2-1.	Components of a Case-Based Planning System	15
Figure 2-2.	Enhanced Expert System with an External Database	18
Figure 3-1.	System Data Flow and Data Generation Sequence	27
Figure 4-1.	Integration of Manufacturing Functions within the Centre of Relational Databases	37
Figure 4-2.	The Logic Structure of the Database System:- The Straightforward Process of a Forging	38
Figure 4-3.	Database Structure for which a Versional Strategy is Required	41
Figure 4-4.	Menu System for Metallurgical Functions	42
Figure 4-5.	Database Structure for Multiple Forging Processing without Versions	45
Figure 4-6.	Pointers in Forging Status File to link Single FEL Order No. With its Bundles in Forging Definition File	50
Figure 4-7.	Manufacturing Plan Data Drawn from Different Relations	59
Figure 4-8.	Manufacturing Plan Relations Hierarchy	60
Figure 4-9.	Internal Relationships between Versions for some of Process Plans in Quality Plan Environment	66
Figure 4-10.	Manufacturing Plan & Quality Plan Relations Hierarchy	67
Figure 4-10a	ER Diagram for the Manuf. Plan & Quality Plan Relations	67a
Figure 4-11.	Menu System for Quality Assurance Functions	72
Figure 4-12.	Menu System for Test Results Functions	80
Figure 4-13.	Process Planning Logic for FEL Melting Range Design	86
Figure 4-14.	Overlap Storage Structure between Specifications and Melting Ranges	90
Figure 5-1.	Database System Enhanced with an Expert System	99
Figure 5-2.	Gurney-Lurie Chart for Infinite Components	111

Figure 5-3.	N_e vs. Relative Surface Resistance (m) for Single Geometric Shapes	113
Figure 5-4.	Long Cylinders: Variation of Temperature with Position for Different Values of Relative Surface Resistance	115
Figure 5-5.	Heat Treatment Cycles for Component going through Different Temperature Stages	117
Figure 5-6.	Enhanced Database System: External Enhancement	120
Figure 6-1.	System Output of Manufacturing Plan	125
Figure 6-2.	System Output of Steel Order	126
Figure 6-3.	Expert System Output of Approved Steelmaking Procedure	129
Figure 6-4.	System Output of Calculated Hydrogen Degassing Hours	130
Figure 7-1.	Knowledge Development Process	139

INDEX OF APPENDICES

APPENDIX 1 Fields, Relations and Their Definitions A1:1 to A1:23

1-1 Fields in the Various Status Files A1:1

1-1-1 Forging Status A1:1
1-1-2 MP_Status A1:1
1-1-3 QP_Status A1:1
1-1-4 SO_Status A1:2

1-2 Fields in the Relations in the Forging Definition Planning Area A1:2

1-2-1 Forging Definition A1:2
1-2-2 Specifications A1:3
1-2-3 Chemical Range A1:3
1-2-4 Mechanical Property A1:4
1-2-5 Customer A1:4

1-3 Fields in the Relations in the Manufacturing Plan Planning Area A1:5

1-3-1 Manufacturing Plan A1:5
1-3-2 MP_Status A1:5
1-3-3 Operations A1:5
1-3-4 Test_Units A1:6
1-3-5 RMC_Scrap A1:6

1-4 Fields in the Relations in the Quality Plan planning Area A1:6

1-4-1 Quality Plans A1:6
1-4-1-1 Quality Plan A1:6
1-4-1-2 QP_Status A1:7
1-4-1-3 QP_Operations A1:7

1-4-2 Forging procedures A1:7
1-4-2-1 FP (Forging Procedure) A1:7
1-4-2-2 FP_Status A1:7
1-4-2-3 FP_Comments A1:8

1-4-3 Heat Treatment Procedures A1:8
1-4-3-1 HTP (Heat Treatment Procedure) A1:8
1-4-3-2 HTP_Status A1:8
1-4-3-3 HTP_Comments A1:9

1-4-4 Clean, Packing and Protection Procedures A1:9
1-4-4-1 CPP (Clean, Packing and Protection Procedure) A1:9
1-4-4-2 CPP_Comments A1:9

1-4-5 Mechanical Testing Procedures A1:10
1-4-5-1 MT (Mechanical Testing Procedure) A1:10
1-4-5-2 MT_Comments A1:10

1-4-6 Special Procedures A1:10
1-4-6-1 SP (Special Procedure) A1:10
1-4-6-2 SP_Comments A1:10

1-4-7 Control Document Status Lists A1:11
1-4-7-1 CDSL (Control Document Status List) A1:11
1-4-7-2 CDSL_Comments A1:11

1-5 Fields in the Relations in the Steel Order Plan Planning Area A1:11

1-5-1 Steel Orders A1:11
1-5-1-1 Steelord A1:11
1-5-1-2 SO_Status A1:12
1-5-1-3 FEL_Melt.Range A1:12

1-5-2 Ingot Requirements A1:13
1-5-2-1 Ingot_Requirements A1:13
1-5-2-2 Ingot_Status A1:13
1-5-2-3 Ingot A1:13

1-6 Fields in the Relations in the Test Results Function Area A1:14

1-6-1 Steelmaking programme from FSL A1:14

1-6-1-1 FSL_Proc. A1:14
1-6-1-2 FSL_Ingot A1:14
1-6-1-3 FSL_Programme A1:15

1-6-2 Cast Analysis from FSL A1:15

1-6-2-1 Analysis_Index A1:15
1-6-2-2 Analysis A1:15

1-6-3 NDT Reports A1:16

1-6-3-1 U/S_Report A1:16
1-6-3-1-1 U/S_Report A1:16
1-6-3-1-2 U/S_Scan A1:17
1-6-3-1-3 NDT_Instrument A1:17
1-6-3-1-4 Probe A1:17

1-6-3-2 MPI Report A1:17
1-6-3-2-1 MPI_Report A1:17

1-6-3-2-2 NDT_Instrument A1:19

1-6-3-3 Dye_Pen_Report A1:19

1-6-3-4 Visual_Report A1:19

1-6-3-5 S_Print_Report A1:20

1-6-4 Mechanical Property Test Results A1:21

1-6-4-1 Certificate A1:21
1-6-4-2 Mech_Tests A1:21

1- 7 Fields in the Relations in the QA Functions Area A1:22

1-7-1 Document Distribution Matrix A1:22

1-7-1-1 Docu_Ditribution A1:22
1-7-1-2 Issue_Status A1:22
1-7-1-3 Distr_Matrix A1:22

1-7-2 Document sent A1:23

1-7-2-1 Docu_Sent A1:23
1-7-2-2 Sent_Status A1:23

APPENDIX 2 Listing of Relationship Definitions A2:1 to A2:12

APPENDIX 3 Listing of Procedures in Individual Menus A3:1 to A3:4

3-1 Procedures in Metallurgical Functions A3:1

3-2 Procedures in Quality Assurance Functions A3:2

3-3 Procedures in Test Results Functions A3:3

APPENDIX 4 Descriptions of Query Procedures A4:1 to A4:87

4-1 Procedures in Metallurgical Functions A4:1

4-1-1 Technical Planning Menu A4:1
4-1-2 Manufacturing Plan Menu A4:4

4-1-3 Steel Order Menu A4:19
4-1-4 Forging Procedure A4:32
4-1-5 Quality Plan Operation A4:34
4-1-6 None Issued QP. Edition A4:38
4-1-7 New Issue QP. Creation A4:38
4-1-8 Quality Plan Printing A4:39
4-1-9 Repeat Previous QP A4:45
4-1-10 Ingot Requirement Menu A4:46

4-2 Procedures in Quality Assurance Functions A4:47

4-2-1 Knowledge Data Bank A4:47
4-2-2 Document Distribution A4:58
4-2-3 Listing Docu. Distributions A4:59
4-2-4 FEL Melting Programmes A4:60
4-2-5 Chemical Analysis Menu A4:72

4-3 Procedures in Test Results Functions A4:78

4-3-1 Test Certificate Menu A4:78
4-3-2 NDT Report Menu A4:82
4-3-2-1 View/Edit NDT Reports A4:82
4-3-2-2 NDT Report Printing A4:83
4-3-2-3 NDT Procedure Menu A4:86

APPENDIX 5 Listing of Exert System Rules A5:1 to A5:23

APPENDIX 6 Hydrogen Degassing Programme in C^{++} A6:1 to A6:4

CHAPTER 1

INTRODUCTION

The development of Computer Aided Process Planning (CAPP) systems is of current interest because it is seen as the principle bottle neck [T89] in the development of systems for Computer Integrated Manufacturing (CIM) [JR87]. In spite of the tremendous efforts that have been made in this development, there are not many CAPP systems in use today. Those that are are concerned with the relatively simple manufacturing processes that determine the shape of components, such as machining [BH87] [CW84] and bending [IK87], and with the processes of assembly [VK86] by which a number of different parts are brought together to make a more complex component.

Manufacturing is, however, also concerned with the internal microstructure, both bulk and surface, of the materials from which artifacts are made. Responsibility for the control of the development of this microstructure is the principal contribution that the materials engineer makes to the control of manufacturing activity. Many different operations are involved and wide consideration must be given to chemical, thermal and thermo-mechanical factors in the manufacturing processes since these affect the microstructure that is developed. Such factors greatly complicate the constraints placed on process planning and this is perhaps one of the reasons why so little progress has been made in the development of relevant CAPP systems. The work described here investigates factors involved in using the semi-generative CAPP approach to support the process planning decisions made by metallurgical engineers in one of the world's leading manufacturers of large steel forgings.

A process planning system to model a manufacturing environment within an organisation should cope with the demanding requirements of CIM [JR87] [CT87] to integrate the development of product planning with the design and planning of the manufacturing processes. This requires that the CAPP system itself is an integrated system [HL88] [HW91] in order to eliminate barriers to the dissemination of information. Tedious repetitive data input will not then be required and errors, previously common, resulting from frequent data input will be avoided [N87].

A database system consists of the stored data (often called the database state) and of meta knowledge about them (the database schema). This research work is based on a relational frame work in which the database state is conceptualised as a collection of tables or relations. Columns of the tables are called attributes, rows are called tupples or relation elements.

This thesis discusses these issues, requirements and features of such a system. The metallurgical CAPP system presented here has been designed and partially implemented in a multiple forging manufacturing environment. The system deals with problems involved in the generation of individual sub-plan together with their integration into the final results. On the basis of a relational database structure with built-in advanced query facilities, it supports the entire metallurgical process planning design and manufacturing sequences. All versions of the metallurgical process plans are properly maintained in terms of access, execution and archive.

This thesis describes the relational database's design used to implement this metallurgical CAPP system in the large steel forging manufacturing environment. The aim is to automate computer support given to metallurgical process planning in this manufacturing environment, and to demonstrate the capabilities achieved in the system. The system is composed of several different routines each undertaking a specific task. Various plans can be generated, the metallurgical engineers being assisted by the database system that has been developed. Substantial facility is provided to retrieve previous experiences or standard proformas.

The process plans and the manufacturing task activities are modelled into a two-level hierarchy. A large amount of data is handled, including customer requirements, approved and developing process designs, test results, information about QA activity and manufacturing knowledge accumulated within the organisation. The plans and the final results are integrated into a single system together with the QA activity.

Manufacturing rules are classified and separated from procedures that actually use them. Rules are stored in the form of rule templates. Users can modify these rules as needed if any process has been changed. By using query sub-system to control and manipulate rules and data in the relations, the system can gain and represent engineering knowledge. Examples are given to demonstrate system learning and representation ability, and the ability to solve diagnosis type problem in such an engineering domain.

On the basis of the database architecture, further development was aimed at system enhancement. Two enhancements have been explored: the enhancement with a small expert system and with an external program. In the first instance, rules concerned with the approval of steelmaking procedures which were stored in data files as rule templates with approved values, have been incorporated into a rule-based expert system using the CRYSTAL expert system shell. Factors such as the speed of operation, storage space requirement, and efficiency in supporting the decision making process were examined in comparison with the use of rule templates. The enhancement with an external program has been tested by automating the process planning of hydrogen degassing. This planning process involves consideration of the final required hydrogen levels, the hydrogen levels achieved by the steel making process, and the hydrogen degassing effect of annealing steps specified for forging and micro structural development purposes. The system enhancement is based on fundamental principles governing hydrogen dissolution and diffusion rather than on the rules developed from operating experience. Issues raised by the enhancement that relate to overall control are then discussed.

This thesis presents an investigation for the design of a computer aided process planning system for Forgemaster Engineering Ltd (FEL). The system designed, called FEL CAPP System, combines a variant approach and a generative approach. This approach makes use of a dynamic knowledge base containing old cases, i.e. part specifications and corresponding process plans. The knowledge base grows with experience and will increase planning quality in the future. Furthermore, it models real-time process planning logic and enables an automated improvement of similarity metrics which can be regarded as learning. As will be seen, the system provides a short-term solution to current problems, but remains adaptable to long-term business requirements. It covers a wide range of manufacturing activities and demonstrates the ability to capture manufacturing methods and data. Problems associated with the implementation of the system in manufacturing are also highlighted.

CHAPTER 2

LITERATURE SURVEY

2.1 Process Planning in Manufacturing

The main function of process planning within a manufacturing organisation is to select the manufacturing processes, and other activities, which will be used to transfer a part from its initial state to the final state determined by customer requirements. SME defines process planning as the systematic determination of the methods by which a product is to be manufactured, economically and competitively [T85]. Chang and Wysk (1985) state that for a computer-aided process planning system (CAPP), it is necessary to define part design precisely, identify and capture the logic of process planning, and incorporate them in a unified manufacturing database.

Process plans, themselves, specify the appropriate sequences of production operations and the necessary tools and facilities to ensure the right manufacturing practice being adopted. It is absolutely necessary that effective productive time, output quality, speed, and accuracy should be achieved to meet customers' requirements.

Although computer aided process planning has been active area of research over the last two decades, for the following reasons process planning is still difficult to automate:

* Finding a complete part representation (and interface to CAD systems) is essential to generate correct process plans. It must contain not only geometric but also technological information on different levels of abstraction in order to support different planning tasks.

* Process planning is characterised by strong interactions between sub tasks which may be resolved successfully by human experts. Up to now, no coherent body of theory has been developed for process planning.

* Given the dynamic nature of manufacturing technologies, the system must support updating of knowledge to guarantee plans with high quality over a long-time period.

Traditionally, this problem has been solved by planning engineers manually producing process procedures. Normally they are hand-written process plans. This task involves examining an engineering part drawing and developing process plans and instructions based upon the process and machine capabilities, tooling, materials, related costs, and shop practices.

The deficiency of the traditional approach is that engineers, whose skill and experience could be deployed solving manufacturing problems, spend much of their time in a purely clerical role [SE82]. Such clerical effort involved in manual production routing generation, followed by job time determination, document production and finally operator inspired time 'adjustment' is simply unacceptable in the current competitive market. However the overriding argument concerns manpower availability. This has two facets. The ongoing shortage of skilled operators on the shop floor has long been recognised; The second is the population in the process planning department itself close to retirement, who will take with them a wealth of irreplaceable manufacturing knowledge related to a specific manufacturing organisations.

Another important factor of CAPP, which has not been recognised completely in the past research, is that it forms part of Manufacturing Information System. It is generally concerned with computerised information to support not only the process planning domain but also a company's business and manufacturing strategies. Unless such information is stored, processed and fed back as part of an overall company operations plan, CIM will not be a reality [G88].

2.2 CAPP System Approaches

There are many CAPP systems in existence today. Chang and Wysk (1985) and Alting and Zhang (1989) contain surveys on existing CAPP systems. It is worth noting that not all CAPP systems are the same. Indeed, issues related to the development of CAPP systems can be viewed in terms of the traditional classification into three types: variant systems, generative systems, and, a mixture between the other two, the semi-generative systems. Artificial Intelligence/Expert Knowledge can facilitate the implementation of generative process planning be alleviating the time-consuming and labour-intensive chore of modelling each and every unique situation.

2.2.1 Variant Approach

The variant approach has been widely used in CAPP. The generation of a plan involves identifying, recalling, and retrieving an existing plan for the same or a similar part. The recalled plan is either a historical plan or a pre-defined master plan and, once recalled, is modified by the process planner to satisfy the current requirements.

Group Technology (GT) has been successfully used in this type of CAPP. Parts are grouped into families based on the similarities of features of a part, including its geometric shape, size and processing requirements [JV88][NC85]. Standard process plans are manually defined and stored in a computer database for each family of parts. When a new part is introduced into the manufacturing system, it is first classified and coded. The code is used to retrieve a similar plan from the database. CAPP can retrieve these plans in some sequential fashion and output a variety of documents.

This is more like human activity but, using the computer as an assistant tool shows a dramatic improvement in information organising capabilities, and reduce the time and labour consumed. The new plan is restricted to the historical performance and therefore needs the involvement of an experienced planner at the final stage to make it fit the new requirements. The variant approach, however, is still a practical way for a manufacturing organisation to produce plans of the operations to be undertaken. For single job environments, it takes less time to develop a variant plan using a system than a generative one. At the current stage of CAPP development, the application of the generative approach to a multiple job environment could not be considered viable.

Systems entitled to be classified under the variant approach include CAPP-CAM-1 [L76], MIPLAN [S80], A-CAPPS [GG92] and some others. They have been successfully applied in industry and found to be highly successful in increasing the information management capabilities when compared with manual systems. The developed systems take production capability, machine capability, and operation cost into account to determine feasible process plans for a product mix. Examples of systems using such coding approaches include CAPP-CAM-11 [L76] and MIPLAN [S80].

The computer assists by providing an efficient system for data management retrieval, editing and high speed printing of process plans. In most batch manufacturing systems where similar components are produced, variant systems can improve planning. In the domain of our application in which both geometric features and mechanical properties are concerned, such coding approach can certainly help not only retrieving the right previous or master plan, but also achieving precisely the chemical range requirements in order to obtain the desired internal structure of the steel.

2.2.2 Generative Approach

The generative approach aims at full automation of process planning. It has attracted a great deal of attention from researchers [HL88] in the past two decades, probably because of developing shortages [D87] of skilled and experienced process planners in many industries, but also as part of the move [GL87] towards the ultimate goal of unmanned production in the future. In the generative approach, standard manufacturing plans are not pre-defined and stored. Its main function is to synthesise a new plan for each specific part. To do this the system must contain information about the manufacturing rules governing the activities involved, about tool availability and about the capabilities of the equipment to be used. There must also be a built-in logic engine that will allow the computer to use this information and to take processing decisions. Most of the systems use Group Technology combined with decision tables; decision trees [D87]; or other methods to capture the manufacturing logic such as analytical models [B90i] [CY91] or simulation models [Z86].

Although the generative CAPP approach does not involve a process planner, the dialogue mechanism between human and machine - defined as interactive input - or between the CAPP system and CAD system - defined as interface input - is still necessary in order to provide precise geometric and other design information to enable the computer to act. If such interface problems can not be solved, the current systems can not be defined as truly generative systems.

The generative CAPP approach has achieved some success in the planning of individual operations - such as TIPPS [CW84], XCUT [BH87], for machining piece-parts, TOM [M82] for machining holes, EXCAP [DD84] for rotational components, GARI [DL85] and AMPS [IK87] for metal cutting and bending, and many other unnamed or prototype systems. Many of these systems are still in their experimental stages and have never been applied to a real production environment [TK88b]. They are, moreover [AZ89], limited to one operation or to the manufacture of one particular part. But manufacturing is concerned not only with the individual process but more importantly with overall process plans. Thus it involves production planning, process planning and operation planning. These activities are highly interrelated in practice, even though each has its own distinctive goals and scope [HL88]. The integration of all these plans with their individual functions within a single system is the ultimate goal of CAPP research.

2.2.3 The Semi-generative Approach

Finally, we may add another classification of process planning - the semi-generative CAPP approach which can be seen as an intermediate step towards fully generative CAPP systems. Semi-generative CAPP works as an advanced variant approach and offers several flexible options to the users, including refining and evaluating incomplete plans. At the moment, the use of fully generative systems is yet to be seen in real industrial manufacturing environments [HL88], so that semi-generative systems retain their competitive strengths.

2.3 Techniques in CAPP Research

This section will review some techniques based on the above approaches from CAPP research, most of which are aimed at using the generative approach to achieve automatic CAPP function. This is not a complete survey, as the research is growing rapidly, but rather its function here is to indicate trends in the research that is underway and to demonstrate the directions in which it is developing.

2.3.1 AI Techniques

Artificial Intelligence (AI) or Intelligent Knowledge Base Systems (IKBS) is another technique in the latest trend of CAPP research that belongs to the generative approach.

AI is included here is because it has had a major impact on the development of generative CAPP systems since the beginning of the 1980's. This impact has been driven by the successful utilisation of AI in other fields, resulting, for example, in the expert systems such as MYCIN [HK85] and NEOMYCIN developed in the mid-1970's for medical diagnosis and consulting. This has resulted in increased numbers of similar systems in which AI techniques have been applied [DD84, TK88a, BD90] [SB88, B86, JF89]. The techniques commonly used for representing process planning knowledge and the characteristics of a component include the following [LL90],

(a) nodes and links in associative or semantic networks;
(b) condition-action pairs in rule-based or production systems; and
(c) slots, facets and values in frame or object systems.

The automation of the process planning function, however, is a difficult task because the required knowledge is highly complex and ill-structured. It is extremely difficult to formalise the decision-making process of a human process planner because of its subjective nature and the overlapping functions which interact with each other. In addition, the knowledge can't be easily represented using a single paradigm of reasoning, but requires an effective blend of knowledge representational techniques [JS90, LL90, LL93]. This has resulted in extensive research into CAPP systems as will be reviewed from section 2.3.2 forward.

Most of these systems, however, are firmly fixed in the process planning for one process or for one particular machined part. Because process planning, even for single part, requires considerable amount of human expertise and manufacturing knowledge, CAPP, now at the beginning of 1990s has not yet shown many successes in spite of all the research and development effort that has been expanded. The AI Systems, or Knowledge Based Systems, in contrast with the systems based on decision tables, are, however, reported to show dynamic and significant improvements in terms of automation and consistency.

In these early developed systems, AI researchers examined the methods used by human experts in the performance of such tasks, ascertained what knowledge they brought to bear, and explicitly encoded much of it into programs. Two major approaches have been taken to encoding knowledge in computer programs: knowledge as rules (of inferences to be made and actions to be taken) and knowledge as frames (structures of knowledge) [KB92]. They were built of hundreds of inter-linked program units. The ability to modify such a system is severely limited. Moreover, it is unlikely that the expert, the process planner, would have the programming background to enable him or her to add more expertise - i.e. to make the system smarter.

Newer systems are based on a new software technology: expert system shells. By using these tools, an expert system can be built up within a knowledge domain very quickly. This type of expert system has great advantages over the early classic expert systems such as MYCIN which were built up incrementally over long periods of time using a discursive programming language such as LISP.

2.3.2 Knowledge Capturing and Representation

The latest trend of research into automated process planning involves the use of IKBS techniques. knowledge is represented primarily as either rules or simple unconditional facts. The so-called knowledge base of an expert system comprises mostly rules plus other sources of information. This information can take on many different forms from the simple to the esoteric. Its purpose is to serve as the raw data to be used by the inference engine to determine the applicability and subsequent truth of rules. In a process planning expert system, this would include information on materials, time standard, tools, instructions, etc. Examples of knowledge include heuristics, rule-of-thumb and expertise.

As their name implies, the expert systems are meant to be particularly good within a limited knowledge domain or area. In effect, they emulate experts. They are not generally intelligent; This type of system has proven very difficult to develop and this difficulty was the prime motivation for the design of expert systems shells. They have no inherent knowledge, except in a particular area of reasoning. To functioning, they require knowledge and data about a specific domain. The most popular and widely used of these shells are the production systems.

A production is a rule that specifies what to do if something is true. A rule usually takes the form of an IF-THEN statement, a premise (also called (pre-)condition, left-hand-side (LHS) situation, if-part, ...) and a conclusion (also called action, consequence, RHS, then-part, ...). Generally, the rule encodes the knowledge that if the premise is true, then the conclusion is also true (or should be acted upon). The THEN part consists of one or more conditions or actions to be performed. These actions might create new information, perform calculations, produce some output, ask the user for input, etc.

A rule in this form, along with others, is the means by which knowledge - expertise - is captured. The number of rules necessary to emulate some desired level of expertise can run anywhere from tens to thousands. Collectively the rules constitute what is known about a particular domain of knowledge, be it deciding how to manufacture a part or perform diagnosis.

Superficially, at least, it would appear that an expert system written using rules is not greatly different from a program written in a procedural computer language that has IF-THEN statements. The similarity is purely synthetic. The underlying difference is in how these rules are used.

The difference is that there need not be any particular order to how rules are written in the expert system shell. Rules can be added, changed and deleted much more easily than similar modifications with a statement in a program.

The current state of the art is that the amount of knowledge in most of the systems built to date is very small in comparison with the amount in the databases used in most organisations, and so far, there are no successful means of taking advantage of large databases in knowledge-based systems [S86]. The need for integrating AI and database technology is seen as an essential step in the development of future information systems.

2.3.3 Case-based Planning

A large variety of computer-aided techniques have been published in the study of CAPP research. Based on Schanks Dynamic memory theory [S82], the method of Case-Based Planning (CBP) [H86] [HS91] has been developed. It is computational method in which knowledge may be inferred directly from past events called cases. Humm et alii [HS91] introduced components of a CBS system (see Figure 2-1): the case base is a memory of old cases which consist of tasks and corresponding solutions. The retriever selects an old case from the case base which ties closest to the current task according to a similarity metric. The modifier adjusts the old case's plan to the current task. Here, the process planning engineer is assisted in constructing process plans and in feeding manufacturing related concerns back to the design process.

This knowledge process allows new process plans to be generated from old experience.

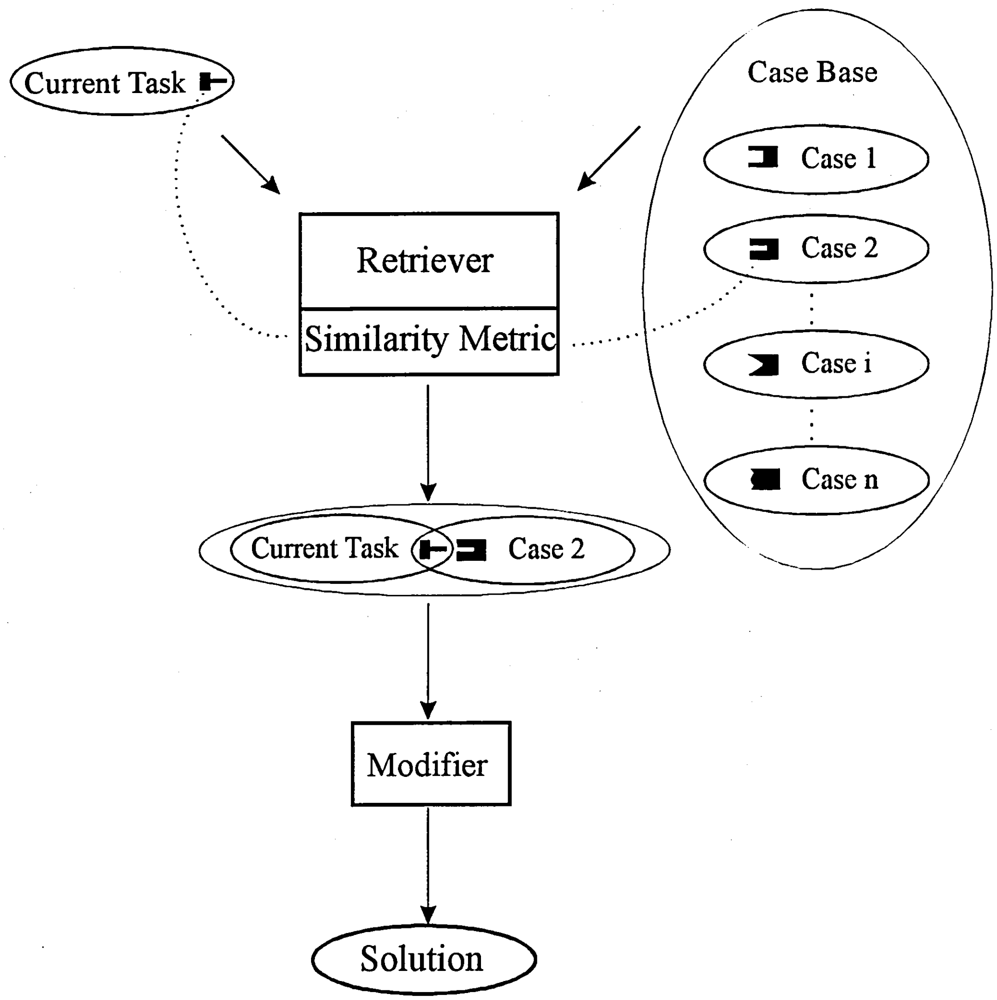


FIG. 2-1 Components of a Case-Based Planning System [HS91]

Therefore, it may improve planning efficiency and plan quality. In the context of the concurrent engineering, such technique provides a flexible and opportunistic style of process planning.

Concurrent engineering (CE) [S92] is a product development strategy which emphasises simultaneous consideration of various product life cycle concerns at early stages in the product development process. Traditionally, design and manufacturing activities have been serially linked by the process planning function. One objective of concurrent engineering is to transform this sequential approach into a more interactive parallel process. Therefore, computer-aided process planning systems must be developed that support an opportunistic style of process planning [HL93]. Obviously, the traditional method, whereby a CAPP system enforces its pre-defined process planning style on the engineer, will not satisfy the CE requirements. What is needed is the CAPP system that should support the engineer's individual planning style and experiential knowledge and not impose control restraints which ultimately inhibit progress. The engineer can be most effective when computer tools augment successful work habits.

2.3.4 Coupling of a Database with an Expert System

Databases system and expert system have their own advantages. The combination of the two technologies would benefit both expert and database system [D85] [AG87] [II86]. There are many discussions in the published literature relating to the future of information systems about inter-relating the two technologies [BM86] [W84][JC89]. One of the major strategies to integrate these two systems is system enhancement. This strategy denotes one of the two systems (ES or DBMS) as the major component. For instance, an ES may be enhanced with a sophisticated data access component [RR88][AT88]. This enhancement is commonly found in the area dealing with very large knowledge bases. Most of the enhancements are focused to the access from ES to

DBMS as shown in Figure 2-2 [F87][K86][MW86]. The DBMS is used only as tools to store data or rules, without any consideration of the required functionality and performance.

Such couplings imply the existence of communication mechanisms between the two systems [B90]. These couplings can be further divided into, depending on the degree of coupling, loose and tight coupling [MW86], corresponding respectively to a "Static" and "Dynamic" use of the communication mechanism.

Loose Coupling

Loose coupling is direct communication between an expert system and a database. Loose coupling of an expert system with an existing DBMS refers to the presence of a communication channel between the two systems which allows the expert system to extract data from the database prior to inferring. An inevitable problem of inconsistency may arise if the data collected from the database is used while the original version of the data is updated.

Tight Coupling

In a tightly coupled system, data are retrieved from the database as and when required during the execution of the expert system. The DBMS however, still acts in the capacity of a slave to the expert system as you can see in the Figure 2-2. This overcomes many of the disadvantages of the loosely coupled approach, but such interaction can cause a severe slowing down of system performance.

Building a knowledge-based expert system is a difficult task that requires many

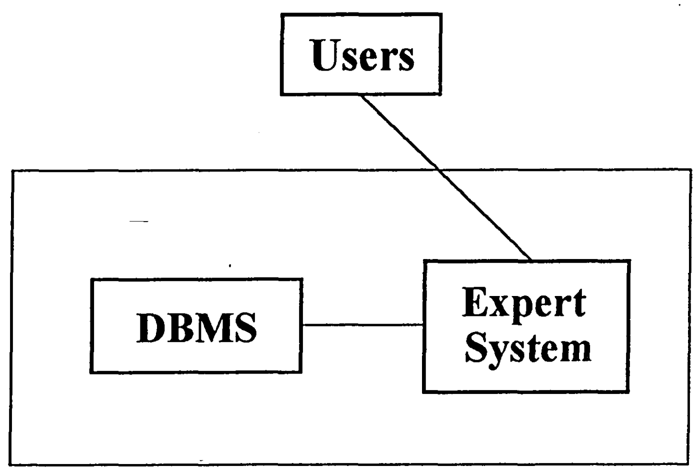


FIG. 2-2 Enhanced Expert System with an External Database

different resources and support. To ease these difficulties, many system development engines (or shells) are developed which hasten the process of building intelligent systems. All of the shells provide basic knowledge representation schemes (e.g. rules, frames, logic) and generic inference methods [K90]. They represent a software environment that is significantly closer to an actual expert system than high level logic computer languages such as LISP and PROLOG. The current wisdom for the creation of an expert system suggests that if a shell is available, it should be used instead of building a new system from scratch.

2.4 Process Planning for Quality

The previous section has discussed the techniques in CAPP research, but the impact of process planning on product quality is the principal concern of most manufacturers. Decisions made in the course of process planning have a significant effect on the resulting product quality, as well as on production time and cost. Because of the extensive research work on CAPP related to metal removal by machining, it is possible for a generic CAPP systems to respond to quality surface finish requirements related and geometric and dimensional accuracy. The manufacturing process of steel melting, forging and heat treatment present a different situation, because they primarily influenced internal quality. Due to the complexity of the relationships between the mechanical properties, microstructure and the production processes that determine them, process planning in this manufacturing domain relies almost entirely on expertise and experience. Planning in this domain cannot be generic but must involve easy access to databases and experiential knowledge.

In addition, computer databases are considered to be the most effective and reliable method of electronic documentation management for both product and process specifications. In today's competitive market, the use of computer databases provides a quality assurance challenge in terms of the tractability and assessment of all process changes during the manufacturing of complex products. Quality control is a manufacturing function which should be integrated as closely as possible into all the operations involved in the whole process, so as to minimise the delay involved in feeding information back to direct the corrective actions needed [RI92]. The integration of manufacturing process planning systems with quality control system will close the quality assurance loop [RI92][W93] and manufacturing enterprises to achieve the highest quality standards

2.5 Summary

In the past few years, an increased number of knowledge based process planning systems has been developed. These systems utilise expert system techniques to represent the knowledge and experience of process planners in a knowledge-based computerised system to establish a fully automated CAPP system. The resulting expert process planning systems are, however, highly domain specific and most have been for manufacturing by metal removal, particularly NC machining. In these manufacturing environments, all components have specified shapes and orientations and a high proportion of the decisions are geometric in nature. That is why so much effort has been devoted to bridging the gap between CAD and CAPP in order to utilise Auto CAD database as the information source for product data [AC93] [FC93]. Process planning within a manufacturing facility that operates manufacturing processes to meet customer requirements, not only for geometry surface finish but also for internal structure is much more difficult to achieve.

Since the task of process planning is experience and knowledge oriented, a CAPP system could be developed by storing the expertise of process planners into a knowledge base which can generate process plans automatically. This is very difficult to achieve, not only because of the complex interactions between process planning and other activities in a manufacturing enterprise, but also because of the distinctive challenge of planning with people. People can cope with the constantly changing demands stemming from product development practice. If a computer system can help people develop this talent, and store and represent knowledge for the future, the reality of CAPP system in industry will certainly change.

Most of the existing CAPP systems actually do operation planning. Their emphasis is placed on selection and ordering. Typically, the emphasis is on issues such as the various ways to make one individual feature (e.g. making a hole by spade drilling or by twist drilling) and the selection of the optimum method. These operations are shape oriented, a CAD system is principally involved. The systems use automatic feature recognition to interpret a part design so that process plans, fixture designs, GT codes and numerical control (NC) programs can be automatically generated [LL93, W91, W92, KB92].

Although process planning systems have been developed for the assembly of electronic printed circuit boards [SL93], process planning with a view to achieve internal material structure and required mechanical properties has not been attempted. Principles of planning are lost in a large body of rules that are difficult to update in a consistent manner [LL93].

The principle feature of the system developed in this work is that the planning knowledge involved is based on experience and is organised into an established structure. Unlike most of the existing Artificial Intelligence (AI) based process planning systems, this approach makes use of a dynamic knowledge database containing old cases, i.e. part descriptions and corresponding process plans. Search algorithms select the 'best' process plans from the historical records. Process plans are generated and subsequently optimised by users. Such procedures can be made more efficient by the use of computers, leading to advantages such as increased planning quality with concurrent reduction of cost of planning. This enables successful process plans to be repeated for new orders.

CHAPTER 3
PROJECT BACKGROUND AND TECHNIQUES

Considering the limitation of generative CAPP in the scope of activities that the rules can cover and in the realisable effectiveness of such systems, a variant type of approach has been adopted in developing the system described here. Under this approach, the plan generation process has been separated from plan refining activities. The system involves user modification with semi-automatic evaluation and error detection using rules, formulae, and decision logic. In some generative systems the manufacturing knowledge base is clustered by the system designer into a static and inflexible partitions. In the system described here, on the other hand, a relational database is used to store the information intelligently. The system was first of all created to be likened to a variant CAPP systems, it was then extended with external enhancement to automate some of the process plans. Nevertheless, the metallurgical engineers play the key role in problem solving within the system and computers are used as tools to assist the engineers by doing the jobs requested of them.

3.1 Manufacturing Domain

The system has been developed within Forgemasters Engineering Ltd. - abbreviated from now on as FEL. FEL produces a wide variety of forged components, comprising machined and heat treated forgings and blooms, from carbon, alloy and stainless steel ingots of up to 230 tonnes in weight.

The manufacture of these forgings is a complex activity. The final forged products require good surface finish, tight tolerances and specified internal structure and chemical and mechanical properties. The manufacturing process commences with arc furnace melting of the steel and its refining, partly in the arc furnace itself and partly in a separate ladle furnace and vacuum degassing units such as VAD (Vacuum Argon Degassing), where VCD (Vacuum Carbon Deoxidisation) is carried out for achieving higher purity of steel. This steel is then cast into ingots from which one or more forgings will be made, although some very large forgings may require steel from more than one arc furnace charge to be cast into a single ingot. These steel making and casting processes are carried out in a sister company - Forgemasters Steels Ltd. - but controlled by FEL through a 'steel order' drawn up by FEL's metallurgical engineers.

Once the ingots are cast, they are delivered to FEL for forging, heat treatment and machining. Although the forging process must produce a rough shape close to the final design shape, the metallurgical interest in the forging process stems from the effect it has on the internal structure of the ingot. The forging sequence must involve sufficient forging reduction to break up the cast structure and to close any central porosity that remains after casting.

The forgings are heat treated at a number of stages during their manufacture, depending on their size, to the final structure and shape required. Although the bulk of the heat treatment cycles involved are designed to produce specific changes to the microstructure of the forging, the high temperature holding times entailed provide sufficient mobility to the hydrogen dissolved within the forging for it to diffuse out into the atmosphere. Since the precipitation of dissolved hydrogen during service can be major contributor to the cracking of forged components, this diffusion is of considerable importance. Depending on the hydrogen content when cast, each forging requires a specific holding time at elevated temperature to reduce its hydrogen content to an acceptable level (solid state hydrogen degassing). Hydrogen anneals can thus be required in addition to the microstructure forming heat treatments so that the totality of the high temperature holding times meets the specific hydrogen degassing requirement.

All the process plans, whether they be manufacturing plans, steel order plans or any other plans, are designed to satisfy the steel specifications stipulated by the customer.

The knowledge of how to do this is controlled to the work of metallurgists at FEL. In order to assist in the designing of the process plans, specifications are constructed in three parts: the chemical range, the mechanical property range and standard testing procedures. These are then used as the guide for the building of the process plans.

In this way, complicated domain knowledge can be treated in a rational fashion. Any given steel composition, herein classified as a chemical range, may correspond to more than one range of mechanical properties. For example, a given chemical range, to meet a specification issue "A", would give a certain set of mechanical properties under the heat treatment adopted in one of the process plans. Another customer, however may require the same chemical range with different mechanical properties. In this case, the same chemical range would be defined as another issue of the chemical range specification with an issue letter "B" and then related to the new range of mechanical properties and a different set of heat treatments.

After the process planners have put all the information into the process plans, their knowledge is merged into the resulting plan. Thus the relationships developed between the specifications and the process plans need to be built up in the database so as to assist the planners to design the plans consistently and rapidly. For previously used specifications, the system must be able to access the knowledge already stored in the database about planning to that specification.

3.2 Two-level Hierarchy Control

To control the manufacture processes involved in meeting a specification, several process planning documents are currently carefully drawn up and the manufacturing processes monitored against them. These documents are drawn up either for single individual forgings or for a groups of forgings (multiple forgings) ordered by a single customer even though differences in required geometric shape and mechanical properties may result in slight differences in the manufacturing processes to which individual forgings are subjected.

Chang and Wysk [CW84] analyse the creation of a manufacturing plan as a hierarchical activity comprising 3 stages:

* Production Planning;
* Process Planning;
* Operation Planning.

In terms of this analysis, the overall manufacturing, or quality plan, drawn up for a forging corresponds to the production planning stage and is the activity carried out at the highest level. The other plans or instructions that stem from these two, such as the steel order or mechanical testing plan, can be seen as existing at the intermediate level. In general we will, hereafter, call these plans process plans and use this hierarchy as the basis for our analysis. The Metallurgical Engineers working within FEL, and using this FEL Technical Database, are not concerned with planning at the manufacturing operation level. The interrelation between these plans within this manufacturing hierarchy is shown in Fig. 3-1. Each box represents an entity of interest in the process planning system. It follows the actual principles of the manufacturing processes. The lines joining the boxes are to show the flow of data. The system has two levels with regard to the integration of all the process planning and relevant decision making points. The use of the query sub-system associated with the relational database, allows the entire system to be well controlled and maintained, the decision making process proceeding in sequence from the top to the bottom.

From Fig. 3-1, we can see that the system includes various planning areas, for example,

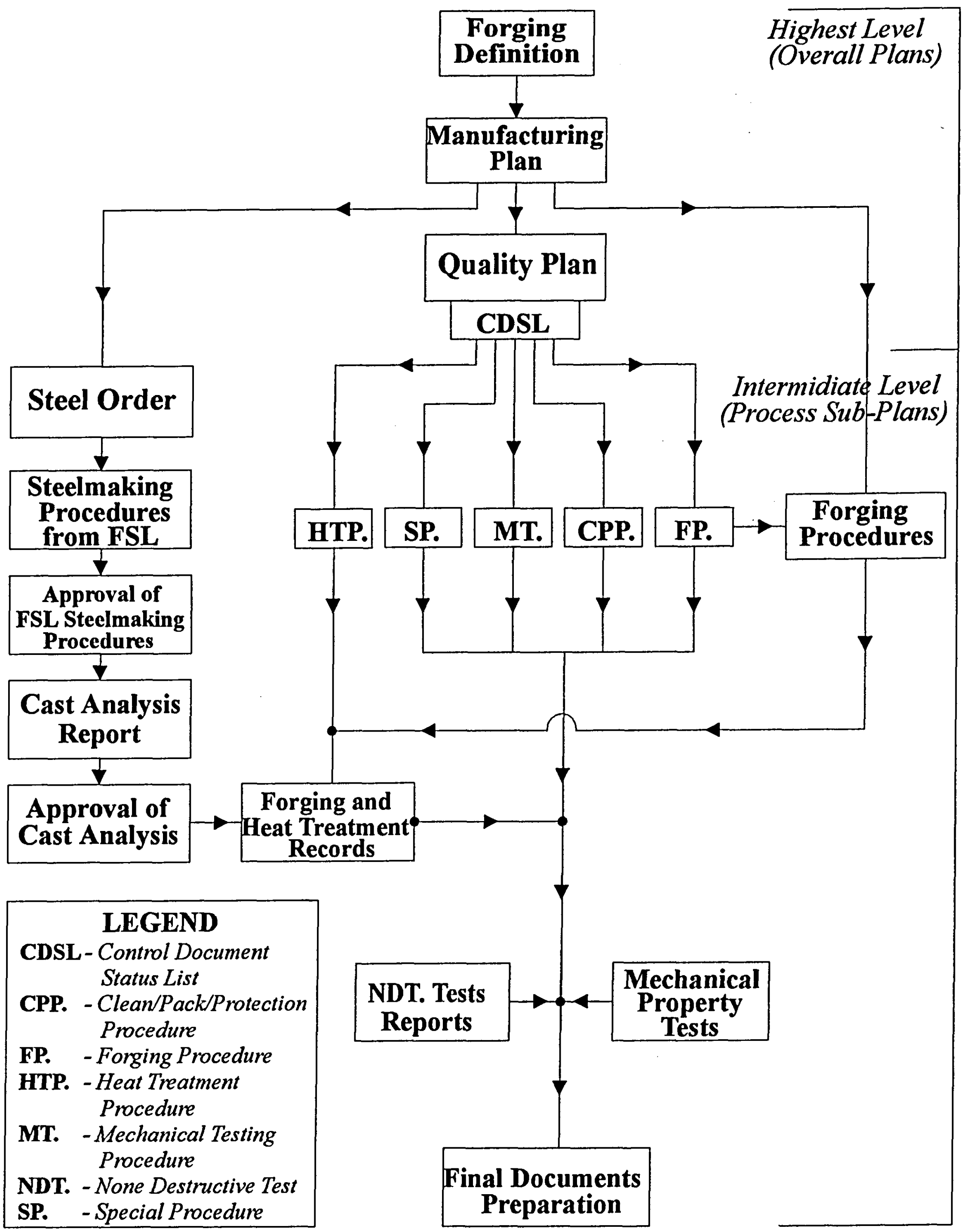


FIG. 3-1 System Data Flow and Data Generation Sequence

for the Manufacturing Plan, for the Steel Order, for the Quality Plan and for test procedures. The operation of such a system requires the following characteristics:

1. A mechanism for the creation of individual plans which is flexible in use and allows plans to be generated quickly and consistently.

2. The integration of all the plans within the system, so that plans at the lower level follow procedures determined by plans at the higher level, and plans at the higher level are supported by plans at the lower level.

3. The ability to maintain metallurgical process knowledge within the system for the support of plan generation, together with the ability to add new knowledge as it is developed.

4. The monitoring of plans under development to detect design errors.

5. Protection of entered data against accidental modification or deletion.

The use of a relational database allows large amount of information to be available to the users at different levels within the manufacturing hierarchy. Several different versions of each process plan may, however, exist at different times within the multiple user computer system necessary to support the hierarchy. The data stored and available within the database must, therefore, be partitioned according to the needs of the user to whom it is available and must be relevant to the appropriate state of the plan to which it relates - data used during plan design being partitioned from data used during plan execution.

Hierarchical control of this nature has been adopted in many industrial control applications for over the last decade [AB85]. It provides a natural way to produce intelligent automatic responses between the different levels [SH82]. As stated before, the FEL Technical Database is concerned with planning at the upper two levels. Plans at intermediate level can be generated according to the requirements specified at the highest level. On this level, the specifications, operations and other activities are specified, on the intermediate level, individual operations and activities are defined. The lower level, which is not involved in the FEL's technical Database, is concerned with the operation of items of plant, for example, to perform the forging reductions, heat treatment cycles etc. specified at the intermediate level.

3.3 Relational Database Systems

Chang and Wysk [CW85] state that for a computer aided process planning system, it is necessary to define part design precisely, identify and capture the logic of the process planning, and incorporate them both into a unified manufacturing database.

A database is defined as a collection of structured data that can be shared between different parts of an organisations information system. Relational database systems, developed from structure models, organise data in single uniform representations. Items in a relational database are represented in the form of two-dimensional tables related together by common attributes. The rows of such tables are generally referred to as tupples, this term having a more precise definition than row or record. Columns are usually referred to as the attributes of the relations.

The applied domain of process planning system has a very complex data structure and therefore requires a high volume of data to be handled. The historical files could be enormous, particularly the files storing engineering knowledge rules and data. The ability to handle such large volumes of data as well as features such as the number of files that can be opened at the same time are typical factor that determine a suitable database package.

More important factor is the relational engine which should allow a database to be built in which relations are open ended. This is important for future development and extension.

When choosing PC based database engine, dBase-IV and DataEase were mainly two competent ones in the market, which can be manipulated under Novell Network. All these DBMSs offer the higher database standard, which is the Structured Query Language (SQL - was originally a standard for mini-computers and mainframes). It was decided to choose DataEase due to distinct advantages it had [CS92]: more reports and forms per database and more records per form, together with multiform access. DataEase allows more than 10 databases to be opened at the same time. The maximum number of file records can be up to 2 billion. DataEase has an advanced query language called DQL. This is made use of in selection processes for reports; in selecting smaller parts of forms; and where a particular field on a form has some processing instructions. The query language has developed into a full programming language, with loops, selects and so on. DataEase also allows menus to be defined for operating the database system. All of these features combine to make DataEase one of the most powerful and flexible database management programs on the market.

3.4 The AI-Database Approach

Knowledge can be represented as both facts and rules. Facts are frequently called declarative knowledge because they declare the relationships between objects. Rules are frequently called procedural knowledge because they define the process by which new facts are generated from old facts.[D91]

Database management systems are engines for handling large collections of facts. Rules can be represented but only in a declarative manner. For instance, report-to-rules can be represented by storing values in a table which records the reporting relationship. Doing this for each such rule in an organisation, however, would mean having tables to represent all possible associations between objects.

Knowledge base systems are traditionally engines for handling large collections of rules. To understand this, it is useful to make the distinction between procedural and declarative knowledge and a procedural and declarative representation for such knowledge. Procedural knowledge is traditionally represented in a procedural manner as high-level language code. What knowledge base systems are attempting to do, however, is in a sense to treat procedural knowledge, like the report-to function, as stored data-i.e., to store it in a declarative representation. This knowledge will be activated by a separate general-purpose processor which will perform all the appropriate inferencing in any particular case.

In recent years the complementary nature of database and knowledge base applications has had led to considerable interest in architecture that combine the two [K86] [GL88] [NH88]. In a database system modelling a real world application, information may be stored in the form of data and rules [R90]. There are two approaches to integrate data and rules in a database management system [DB88] [C89]. First is the homogenous approach. This approach integrates the data manipulation function and deductive features such as inferencing into a single database system. The facts and rules are uniformly represented in a single programming system such as Prolog. In the second approach called the heterogeneous approach [C89], separate identities for the inferencing mechanism and the database management system are maintained and both the systems are coupled through an interface. Both approaches were used for data intensive applications such as process planning in manufacturing.

In this research, the above two approaches have also been adopted together, but there are differences. In the first place, rules are classified and stored in data files as rule templates. These rules can be fired by matching values in a tupple. Therefore, a rule can be separated from a procedure body, and more importantly, users can change these rules from time to time as dictated by technology changes in the future; In the second place, external functions are used to enhance the database query sub-system. complex mathematical problems can be solved through such enhancement.

This approach is better suited for the current application in which standardisation of knowledge development is required. There are two distinct disciplines that need to be balanced when controlling the numbers of approved combinations: for example, the wide and continuously evolving range of applications for forgings manufactured by FEL tends to lead to a proliferation of the range of steel specifications, whereas management and QA pressures are to reduce the range of specifications. From the QA point of view, accurate repeatability of successful manufacturing processes is required; while the diversity of product fosters a wide diversity of process routes.

3.5 Strategies in the Development of the System

An important fact often ignored, is that process planning is a company-oriented task. Manufacturing processes successfully used in the company are not necessarily suitable for producing the same part in another company; because of different manufacturing conditions, like machine tools, skill of the worker, tradition, etc. It is therefore extremely difficult to build a generative CAPP system which can be adapted for use in different manufacturing organisations. By the same token, where a CAPP system is to be developed for use in a particular company, the manufacturing conditions are known, apart from possible future development..

Based on the above discussion, our approach to the development of a practical CAPP system for FEL has been to employ computers as partners in the process planning activity and to work gradually toward the automation of that activity. The decomposition strategy [KB92] has been adopted to break the process planning problem into sub-problems and individual planning areas, each of which can be solved almost independently of the others. Often, solutions in one planning area can affect other activities in other planning areas or functions. The system that we have developed can be classified as a semi-generative CAPP system. because a variant retrieval method and a generative decision logic approach are used. In the system, some decisions are made by the process planning generation logic built-in, while other decisions, which do not need to be generated from "zero", are retrieved in the form of tables, sets of decisions, etc.

The system needs to be compatible with the requirements and nature of the company, due to its potential impact on many activities within the company. Compromise in this area is hard to justify. The system must demonstrate that it can accept the procedures for all the planning stages with adequate control, and that it can meet the form and reporting criteria, the input facilities, etc. that the company operates. Criteria like these can make or break a CAPP system.

CHAPTER 4

DEVELOPMENT WORK ON FEL-CAPP SYSTEM

4.1 Introduction

This chapter describes how a CAPP system has been developed for the manufacturing of large steel forgings using the computer. The applied domain - the process planning system, has a very complex data structure. It therefore requires a very high volume of data to be handled.

The work has been divided into two modules. The first module, which is the database system, forms the basis of the CAPP system that stores all the information in the process planning, quality assurance control and test results functions. The second module consists of the enhancement of the database system which manipulates and control the process planning information. This chapter describes how the computerised CAPP system was developed, whereas, the enhancement is explained in Chapter 5.

A database system has been constructed by modelling the manufacturing hierarchical structure presented in Figure 3-1. The database design requirements are the same as those identified by Rowe and Williams [RW87] where, in designing a database for integrated circuit fabrication, they identified the problems associated with integrating business data and engineering and scientific data within the same database. This chapter of the thesis describes the design of the metallurgical CAPP system at Forgemasters Engineering Ltd, hereafter referred to as the FEL Technical Database. It is composed of several components including a sub-system to assist users in designing the various process plans involved and a control sub-system for the quality assurance functions that maintains the distribution of documents during the manufacturing processing.

The FEL Technical Database system that has been developed uses a uniform method to access data for all the major activities. The data stored can be classified as:

1) Standard information to be shared by the entire system, such as chemical specifications and mechanical property specifications.

2) Knowledge data and rules that are necessary to generate process plans.

3) Manufacturing and process plans residing in incomplete states and in completed (issued) states (the manufacturing plan fulfils Chang and Wysk's [CW85] definition of a production plan.)

4) Forging definitions for individual forging or group of forgings.

4.2 Manufacturing Function Integration

The manufacturing system involves not only the process planning function, but also other functions as well, such as quality assurance, test results, estimation and so on. In the preceding chapters, the importance of integrating the different manufacturing activities has been emphasised. Such integration has three aspects:-

1. shared data, one central database stores the process plan information, with its different versions and supporting engineering knowledge.

2. the user interface with a common style of interaction.

3. communications to pass information between the different computer work stations.

While all three components of integration should be present, the key one is the shared data. This is very important for engineering applications which contain a lot of process plan information. Figure 4-1 has shown an architectural diagram for the integration of the different functions in the system. The central database serves all these functions. This can be illustrated by discussing the typical activities involved in the development of the system.

4.2.1 Simplified Logical Structure

A simplified logical data model for the batch forging manufacturing environment at FEL is shown in Figure 4-2. Each box represents an entity of interest in the process planning system. The name of the entity and its principle relationships are shown in the diagram. The diagram has been simplified to show the logical structure of the main objects, not the way they have been physically implemented. The model has been built around a central data file, named the Forging Status file[1]. The data key to the records in this file is the Works Order Number used within Forgemasters' Engineering Ltd to identify the forging, hereafter designated as the FEL number. Each forging in production within the Company has a separate record in the Forging Status File, and this record provides an essential component of the links between the records related to that forging in the other files in the system. In this large shared data system, each rectangle represents a separate set of records about the manufacture of the forgings, and lines between the rectangles represent the collection of database relations used to model the relationships between real world objects.

4.2.2 Versional Data File Requirements

Figure 4-2 implies that each data file contains a single record for each forging. This is

---

[1] Appendix 1 presents notes on the details of this, and all other files/forms in the database, together with lists and descriptions of the fields in the files.

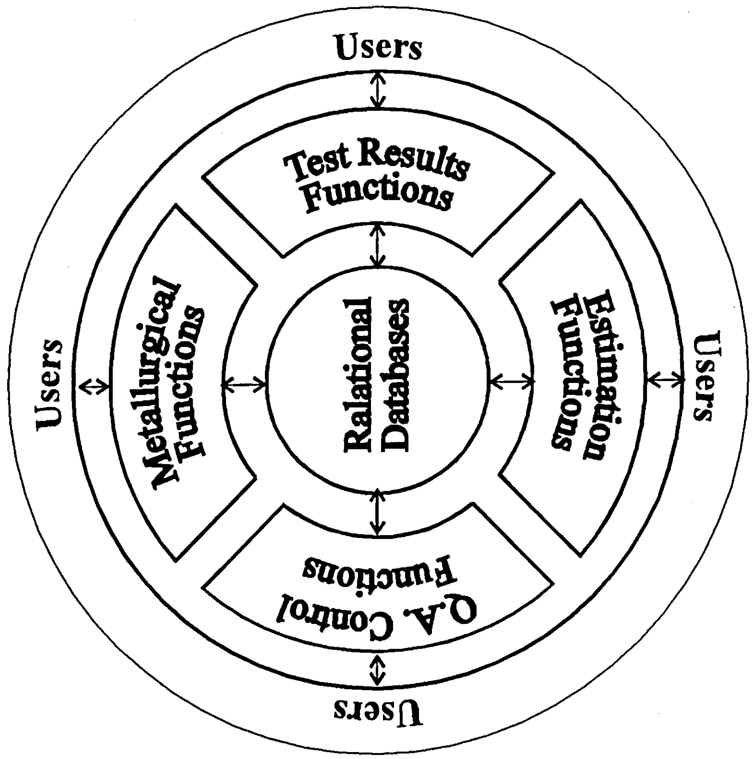


FIG. 4-1 Integration of Manufacturing Functions within the Center of Relational Databases

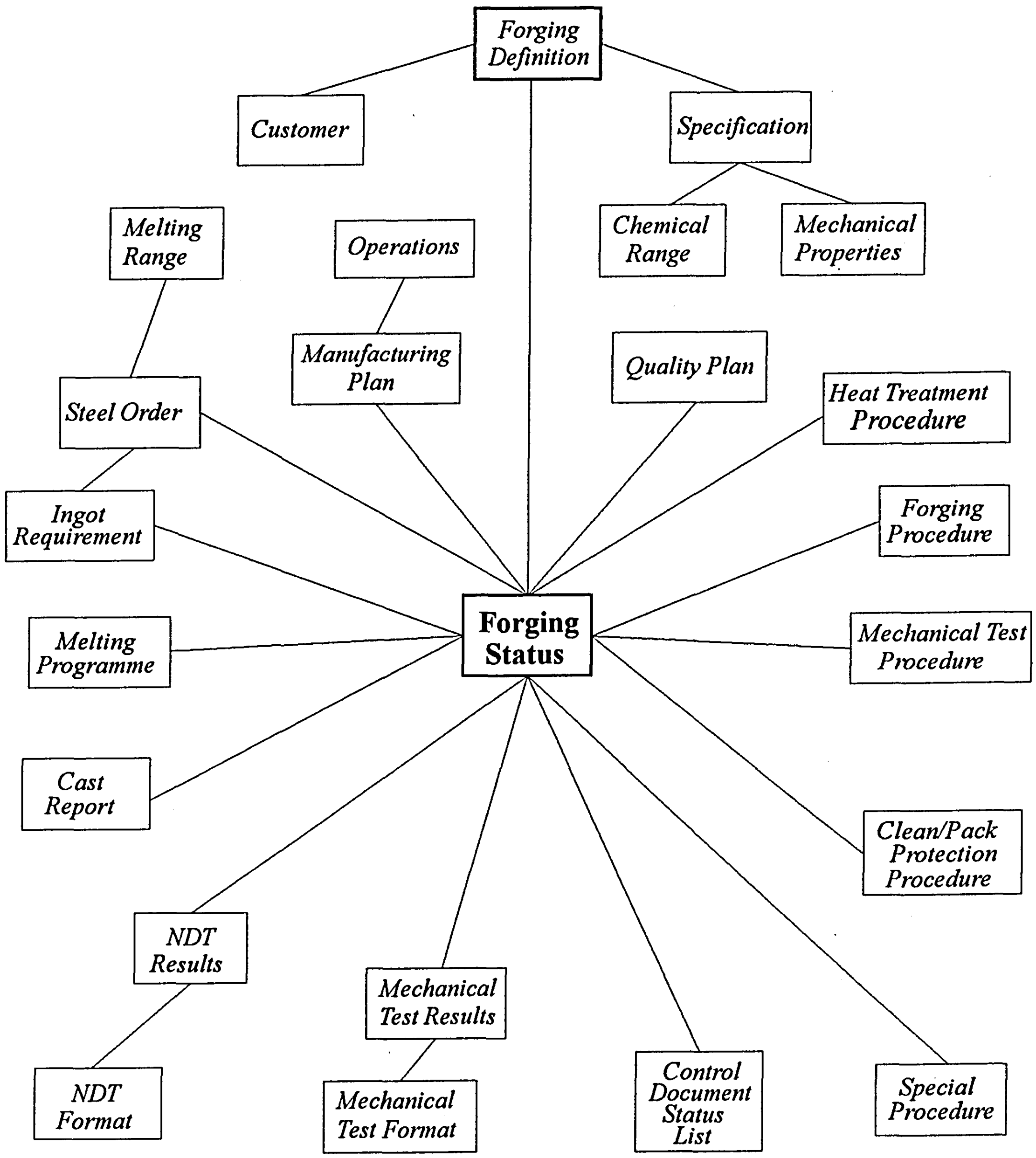


FIG. 4-2 The logical Structure of the Database System:- The Straightforward Processing of a Forging

not the case. Blanken and Ijbema [BI87] pointed out that engineering design is a dynamic activity, going through several different versions before a final version is accepted. A computer system to support engineering design, they argued, must mimic this developmental process. They thus developed the concept of a 'temporal' database able to store different versions of the objects involved. Similar issues have had to be addressed in designing the FEL Technical Database, although the problem is more complicated than that considered by Blanken and Ijbema.

Blanken and Ijbema included a Boolean field "ALIVE" in their data files with a value either TRUE or FALSE. In the FEL Technical Database, the versional information present in a record additionally needs to indicate the extent of the versional changes that the record has undergone. For this reason, each versional data file is provided with a two field key. The first field is an identity number, first assigned in sequence when the record is newly created. This identity number is not changed when the record is revised but the second field component of the record key is a letter that progresses through the alphabet from "A", as the record is changed from version to version. This progression is controlled by the data processing procedures built in to the relevant planning areas in the database, and follows the quality assurance rules laid down within the Company for tracking the revision of, for example, a forging specification or a process plan.

The data processing procedures that create or modify a planning record operate on temporary facsimile files created within the separate planning areas in the system (see section 4.3.1.1), so that draft records can exist locally within that area during planning or modification. Once work or a record is completed, a procedure can be invoked that will issue the record so that further action can be taken on it in other parts of the database. This issuing procedure involves transferring the record from the facsimile file into the real file. If the version letter of the new record is anything other than "A", this procedure will identify the record that is to be superseded and withdraw it to archive.

Data processing procedures throughout the database will not allow records that have been issued in this way to be modified during day to day use. Figure 4-3, with the same logic as Figure 4-2, indicates the data files for which this versional approach has to be implemented.

Further details of the operation of this versional approach will be discussed when the operation of each planning area is described.

4.3 Metallurgical Functions

The metallurgical function is one of the main functions integrated in the current database system, as already shown in Figure 4-1. Activation of this part of the system is through a menu system dedicated to the specialist function in this working area. The menu system is shown in Figure 4-4.

4.3.1 Physical Implementation of Data Structure

The simplest way to establish the relationships that are indicated by the links to the Forging Status file, shown in Figures 4-2 and 4-3, would be for fields in the Forging Status record for a particular forging to contain the keys to those records in the linked files that also relate to that forging. This method is not, however, the ideal method, nor was it the method adopted in implementing the logical structure shown in the Figure 4-2.

Most of the Manufacturing plans and Quality Plans that are designed by FEL's Metallurgical Engineers apply to a group of forgings. If the keys to the relevant linked files were recorded in all the Forging Status records for that group of forgings, substantial redundant data would be present in the database. Danger to the security of

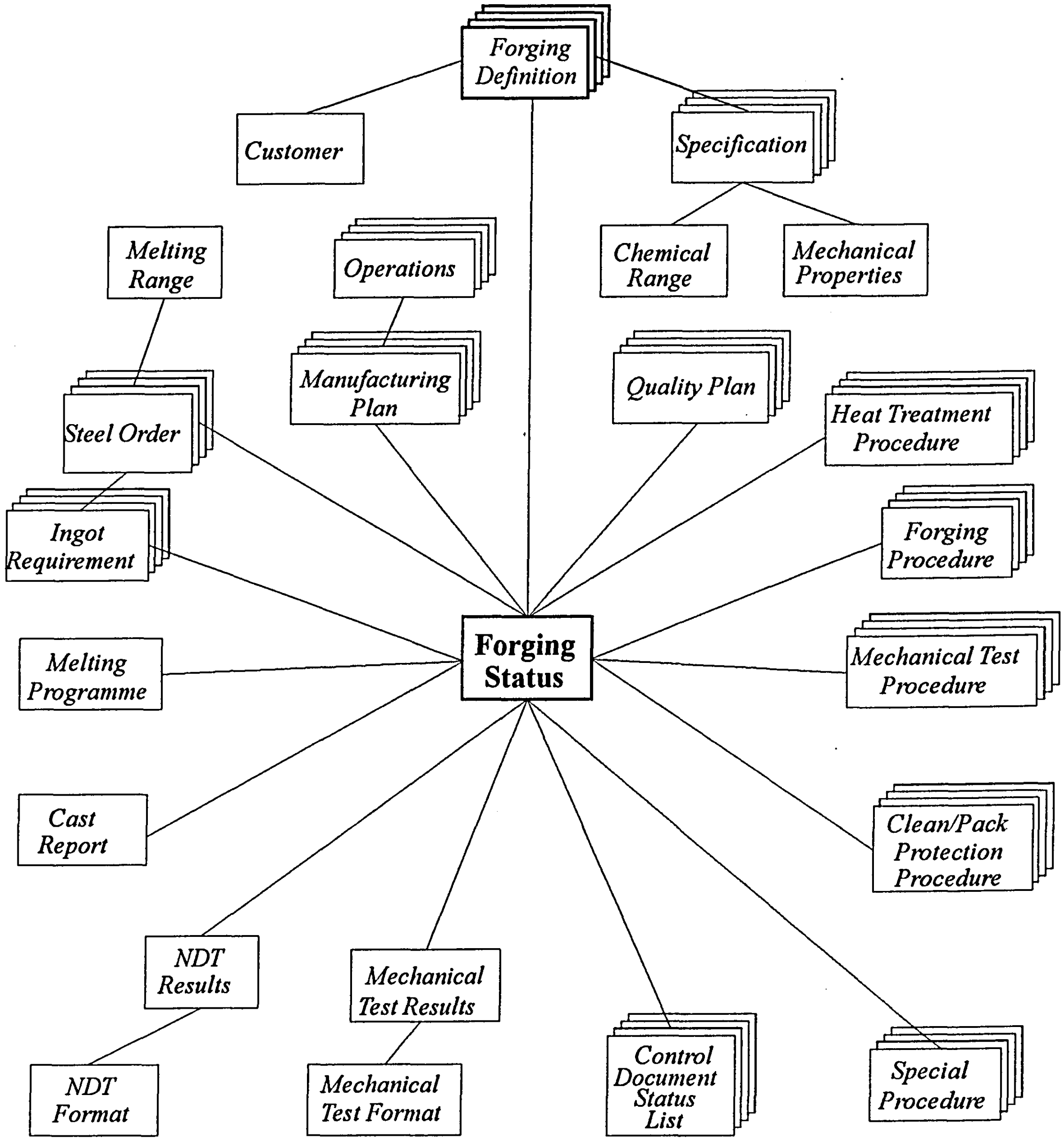


FIG. 4-3 Database Structure for which a Versional Strategy is Required

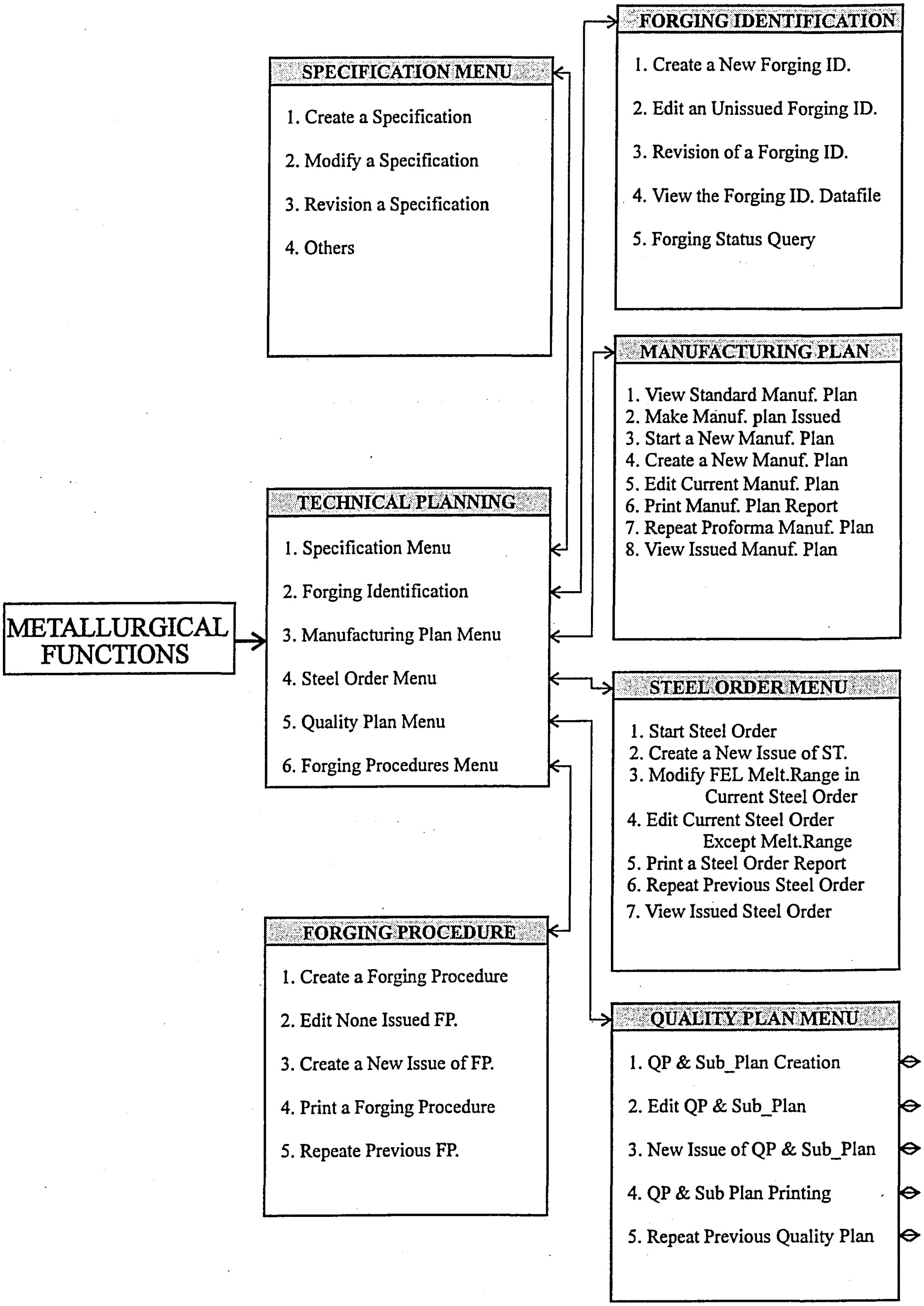


FIG. 4-4 Menu System for Metallurgical Functions (Continued...)

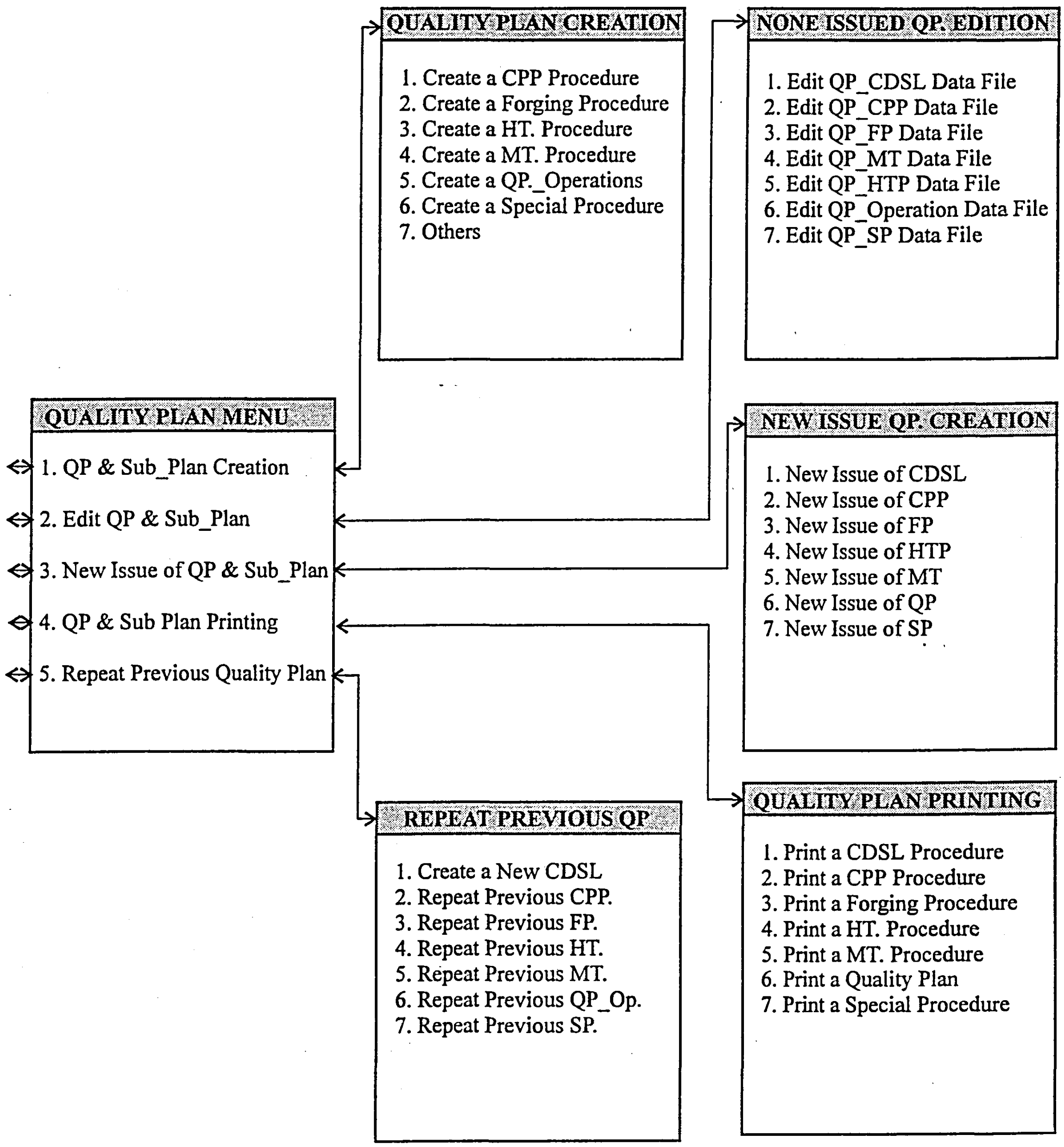


FIG. 4-4 Menu System for Metallurgical Functions

(Continuation)

the database stemming from such redundant data is intensified because a given group of forgings will not necessarily remain as a coherent group between order and delivery. Customers can change the specifications required for a sub-group of forgings after the FEL's Metallurgical Engineers have issued a manufacturing or quality plan for the entire original group. In addition, inaccuracies in the processing of an individual forging at a certain stage in the manufacturing sequence will, if not subject to a concession, require correction through modifications to the requirements placed on a subsequent process. Such modifications require the issue of new versions of the relevant process plans or orders, but only for the forging to which the modification apply.

A further disadvantage of a simple link structure arises because not all the links shown in Figures 4-2 and 4-3 are valid for all the forgings that FEL manufacture. Quality Plans, for example, are only drawn up for forgings above a certain size or if required by the customer. Thus, if the records in the Forging Status file were to have fields available for the keys required by all the links, many of these fields would contain null values for a high proportion of the forgings undergoing manufacture. Null values should be avoided, as DATE has commented [D90], especially for fields that are used for primary attributes. The presence of null value in a field should normally be taken to indicate that the relevant data, or "real world", processing stage has not been completed. Confusion can obviously be generated if a null value in a field could also indicate that the relevant stage is not required.

4.3.1.1 Links between Planning Areas through the Family of Status Files

Figure 4-5 shows the way in which the logical relationships between the different data files were implemented in practice. In order to avoid redundant data and null fields, the facility to chain relationships was utilised allowing the pointers necessary to establish the logical interconnections between the data files to be decomposed into a family of

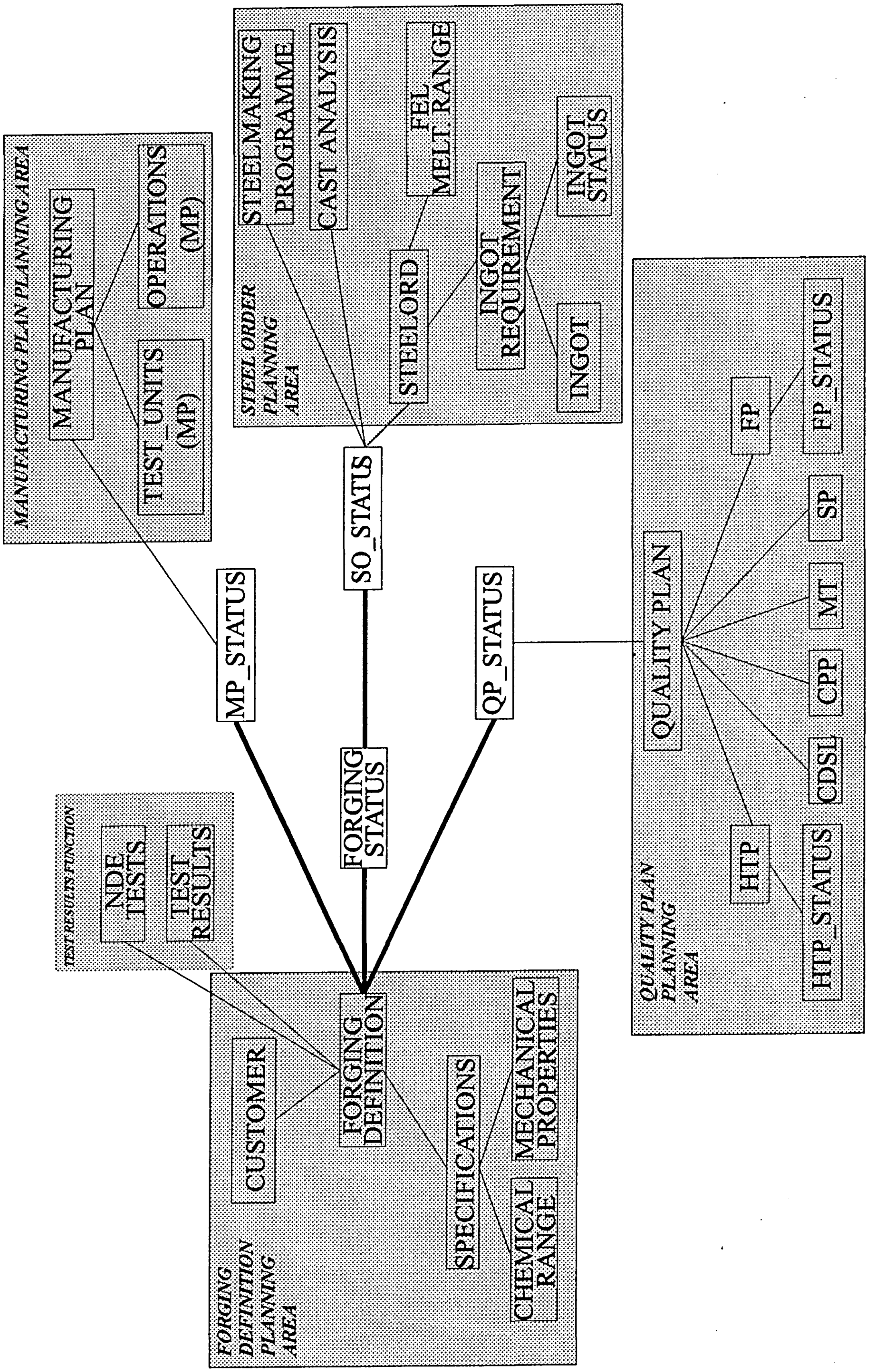


FIG. 4-5 Database Structure for Multiple Forging Processing without Versions

status files. The titles of these status files are Forging Status, Manufacturing Plan Status (MP_Status), Quality Plan Status (QP_Status) and Steel Order Status (SO_Status). Details of these status files definition and functions are listed in the appendix 1-1 together with their relations with other forms. The relationships between these relations are listed in appendix 2 showing how system forms are linked each other.

This structure avoids null fields because, for example, no record will exist in the QP_Status file for a group of forgings if a Quality Plan is not required for that group. The existence of a record in either the MP_Status or QP_Status file always indicates that the corresponding plan should be in existence. The structure also avoids redundant data, since, for example, there will only be one MP_Status record relevant to a group of forgings being manufactured under a single Manufacturing Plan.

Figure 4-5 shows more clearly the four different planning areas within the FEL Technical Database. The Manufacturing Plan planning area, for example, contains three linked data files used to develop and record the basic information for the manufacturing plans. These data files are the Manufacturing Plan file, the MP Operations file and the MP Test Unit file. These files, and the procedures for their manipulation, constitute the Manufacturing Plan planning area. The diagram shows that this area is linked, to the Forging Definition File and then to other planning areas through the MP_Status file.

It can be seen that the status files establish the links between the planning areas. It is necessary to establish and record these links to allow forgings to be grouped in different ways in the four areas. Forgings are grouped in the Forging Definition Planning Area in the way in which they are ordered by customers, individual records in this area applying to these forging groups. Forgings are normally grouped in the same way in the Manufacturing Plan and Quality Plan Planning Areas, but single records are sometimes created in these areas for more than one group from the Forging Definition Planning Area. The function of the MP_Status file, then, is to correlate the keys of the current records in the Forging Definition and Manufacturing Plan Planning areas that apply to the same group of forgings. The QP_Status file discharges the same function as far as the Quality Plan Planning Area is concerned. A separate file is required for this area because, although the same groupings are used initially in planning the Manufacturing and Quality Plans, revisions to plans in these two areas do not necessarily keep in step.

Figure 4-5 shows that the link between the Steel Order and the Forging Definition Planning Areas involves two files, the Forging Status File itself and the SO_Status file. This is because the forging groupings used in the Steel Order Planning Area are determined by the need to cast the steel, from which the forgings are to be manufactured, into individual physical ingots. Depending on the sizes of the forgings to be manufactured, these ingots can be used to make several forgings, a single forging, or large forgings can require steel from as many as three arc furnace charges. The Forging Status file contains an individual record for each forging for which the manufacture process is being planned or is under way, and is the only file in the database that does so. It records the Forging Definition Planning Area key that applies to each forging. The SO_Status file, of the other hand, contains an individual record for each forging for which a steel order is being planned or has been placed, showing the relevant key of the Steel Order Planning Area record. Since some forgings are made for which a steel order is not required, the records in the two files do not remain in register.

Figure 4-5 shows that the collections of the data files and their attendant relations constitute a relational database so that the planning areas are not independent and isolated from each other. The use of the relational database architecture to establish links as required allows data records to be shared by the entire system. Although the family of forging status Data files link planning areas separately to the Forging

Definition file, the facility to chain linking relationships means that these links can be extended to allow the transfer of information directly from planing area to planning area.

4.3.1.2 Functions of the Forging Status File

As stated above, the Forging Status file is the only file that contains an individual record for each forging for which a forging definition has been created and therefore for which an order has been placed. Thus the Forging Status File can take on a monitoring role as far as the Planning processes in all the other areas are concerned. Thus, whenever a new planning record is created in any area, the relevant planning procedure will first check that the FEL order numbers specified for that record exist in the Forging Status File. Planning can only then proceed if the result of that check is positive.

The Forging Status file is also the only file that has highest of security level and can not normally be accessed by system users. Data in it is only manipulated through system query procedures specially designed to perform data entry, modification and deletion automatically, e.g. procedures specified in appendix 4-1-1.

FEL Order numbers provide the key field in the Forging Definition file and in the Forging Status file, but with completely different formats. In the Forging Definition file, a string format is needed to group all the forgings in a given customer order for example eight items from 55432 to 55439; in the Forging Status file, a single number format is required to provide the unique identity of a single forging. As part of the automated procedure that creates a Forging Definition record, the string format is decomposed into the corresponding set of single numbers, and a Forging Status record created for each number, the numbers acting as the key values.

Further procedures were created to update the single FEL Order Number records, whenever a modification of revision was made to a record in Forging Definition file. Thus, each FEL Order Number in the Forging Status relation is always able to point to correct Forging Definition record in order to link other status files together in the manner illustrated in Figure 4-6.

4.3.2 Facsimile File Approach

Before going into detail discussions about each planning area, it is necessary to introduce the concept of facsimile files, which is used to a great extent in the operations described in the next few sections.

Facsimile files are designed as temporary relations identical to sets of records and used for certain of the data process operations. The advantages conformed by the use of facsimile files include:-

1. Performing rules in fields

Rules can be defined to control the entry into a field or apply constraints, for example, validate the control of information. When a Forging Definition record is modified or revision, for example, we want to make sure that all the changes made to the string of FEL Order Numbers are valid. We can build rules to detect whether the new string is within the range of the old string of FEL Order Numbers. This enables subsequent procedure to transfer modified data correctly into both the permanent Forging Status and Forging Definition files.

2. Performing word processing

Unless we use program to rearrange records in relations, DataEase does not allow new records to be inserted among old records. When a manufacturing

Forging Status File

FEL Order No.	ID_No.	ID_Ver.
5567891	00451	A
5567892	00451	A
5553677	00452	A
5567893	00453	A
5567894	00453	A
5567895	00453	A
5567896	00453	A
5567897	00453	A
5567898	00453	A
5567899	00453	A
5553678	00454	B
5553679	00454	B
5553680	00454	B
... ...	... ...	...

Forging Definition File

ID_No.	ID_Ver.	Works_O_No.	Customer_No.	FES	Group	Cust. Drawing	Date_Crd	... ...
00451	A	5567891/2	C1104501	00192	A	TR89235/6	12/07/90	... ...
00452	A	5553677	C1104457	04013	B	TR0864336	01/09/90	... ...
00453	A	5567893/9	C1103387	00932	A	TR8925/31	28/01/91	... ...
00454	A	5553678/9	C1104512	00256	A	TR8927/29	02/05/92	... ...
00454	B	5553678/80	C1104512	00256	C	TR8927/30	18/07/92	... ...
... ...	...	... ...	... ...	...	...	... ...	... ...	... ...

FIG. 4-6 Pointers in Forging Status File to link Single FEL Order No. with its Bundles in Forging Definition File

plan is modified, it is normally because new operations or comments are to be inserted. By using a facsimile file, we can add one more field to indicate the new range the user wants. Once the user is happy with the modified plan, the subsequent procedure can re-arrange the operations placed temporarily in the facsimile file before transferring the relevant data back to the permanent process plan files.

3. Data Security

By linking facsimile files with specific relations, it is possible to limit user's access to certain information in the system. Although DataEase provides a number of security levels which allows a security structure to be designed in the system, it was found not to be sufficient for the complex system developed here in which the knowledge and manufacturing rules are stored.

4. Quarantining Modification Activities

Facsimile files allow the process, by which the issued version of a plan is modified prior to the issue of a revision, to be 'quarantined' from the day to day use of the existing version.

These are the main reasons facsimile files are adopted for the data process operations. Further advantages will be demonstrated in the following discussions in relation to individual problems.

4.3.3 Process Planning and Data Manipulation

This chapter briefly explains the individual planning areas and their data manipulations. Each section will begin by introducing the individual data structures followed by a description of the generation of process plans. In the Steel Order Plan planning section, the generation of the process plans and the development of their corresponding steelmaking programmes will be explained, together with the way in which expert knowledge already stored in the database is re-used in process plans for the same or similar specifications. The discussion demonstrates the capability of database systems to store, search through, revise and extend large amount of data.

Two data processing operations, modification and revision, will be referred to extensively in the report. These have importantly different meanings in this CAPP system. The modification of a record is a process that goes on during the initially planning for drafting stage, before the record has been approved and issued for any action other than planning in the 'home' planning area. During this period, the process planner can make any changes to the plans according to their experience, customer requirements or specifications and standard testing procedures. But once they have been approved or issued for production or for use in any other planning area that issue cannot be changed. This is an essential rule operating within FEL. Changes to an issued record therefore require another data processing operation, named revision.

The revision of a record is based on the previously issued record. When the new record is generated, the operating procedure allocates a new issue letter for it, to distinguish it from the original record. This new record becomes the current one and modifications can be made to it within its home planning area, up to the time that it, in turn, is issued.

4.3.3.1 Forging Definition Planning Area

This planning area develops basic information for use by the whole Database system, much of it essential for the operation of the other parts of the system. Records in this planning area provide the central link between the forging status relations and the information to be shared by the whole system.

There are three main data planning sub-areas, A, B & C below. Each performs individual record processing, creating, modifying and revising operations

The data stored in these relations normally remains un-changed once they have been created. Appendix 1-2 contains a list of the relations in this process planning area, giving their fields and the functions of principle fields.

A. Customer Relation

The customer relation, or file, stores relevant information about all FEL's world wide customers: customer's address, post code, telephone number, Fax number, and so on. Each record is identified by a unique number assigned by the system when a new customer is entered. This number will be used in other relations to draw in customer data as required. For example, each record in the Forging Definition relation needs to have a field, customer number, to link to the data in the Customer relation. A one to many relationship therefore exists between records in the Customer relation and in the Forging Definition relation.

B. Specification Relations

The three relations - Specification, Chemical Range and Mechanical Properties, constitute a data structure for storing the specifications of forgings. The Chemical Range[2] and Mechanical Properties are sub-forms of the specification relation, because each chemical range may have more than one mechanical property range depending on the ingot sizes and heat treatment cycles required by the customers. This structure

[2] The Chemical Range is the maximum and minimum contents of 18 pre-defined solute elements - [C], [S], [O] etc. plus the contents of up to 4 further solutes the can be defined by users.

allows one chemical range to serve several mechanical property ranges to form individual specifications which, in turn, are shared by many other process plans. This minimises the storage space required and avoids redundant data.

Each record in the Chemical Range relation corresponds to one issue of one grade of one specification from a particular source and is automatically allocated an FES (Forgemasters Engineering Specification) code by which it can be identified. A subsidiary relation, mechanical Properties, stores, by Group Letter starting with "A", the mechanical property groups that correspond to each chemical range record. The only data from the Chemical Range relation that is stored in this subsidiary relation is the FES Code.

Record creation, modification and revision are controlled by the specially developed procedures within the database management system. When any one of these operations is initiated, the procedures create one or more temporary facsimile files that can only exist within the planning area. This allows the procedures to check that the local record conforms to specific necessary constraints and quality requirements before it is stored in the database and made available for use throughout the rest of the system. These checks include the followings,

1. When a user wishes to create a new record, the programme only allows the user to proceed if the specification, grade, and issue for the given source do not already exist in the database. This maintains the uniqueness of the records in the database.

2. Day to day users cannot modify any specification which has been quoted in any existing manufacturing plan.

3. The user can only save a new record if certain fields have been filled, primary key fields and foreign key fields, and some required fields, such as specification name, source, and conditions for the heat treatments, these fields being necessary to help users to trace the record.

Users working in the Forging Definition Planning area cannot enter a specification to modify once it has been issued into the system. If further modification is required after the accidental issue of a specification, a formal procedure has to be followed as will be explained in section 4.4.1.1.

C. Forging Definition Relation

As Figure 4-5 shows, the Forging Definition relation plays an essential role throughout the database. The other relations previously considered - the Customer and Specification relations store basic information and need to be created before data can be entered into the Forging Definition relation. This is because the forging Definition relation includes their key fields as foreign keys to link to information they contain.

The Forging Definition relation had no parallel in the manual system that the database system is designed to replace. Most of the data stored in the Forging Definition relation existed in other parts of the manual system. It is necessary to draw it together in the computer database system so as to,

1. provide links between the QP_Status or MP_Status and the Forging Status relation, to which is linked the SO_Status, in such way that the system as a whole is organised to achieve the manufacturing hierarchy structure previously discussed in chapter 3.

2. save disk spaces; the data stored in Forging Definition and its related relations are unlikely to be changed after the works orders have been issued, so a Forging Definition record is not affected by alterations to other process plans. The relations defining these other process plans need only record the Forging Definition primary key values in order to gain access to all the information in the Forging Definition planning area.

3. group multiple works orders together for the same definition. This simplifies the creation, operation and maintenance of the records, and is facilitated by the use of a numeric string data format for the FEL Order numbers,

The rule based query procedures for the creation, modification, and revision of the forging definition records are made available to the user through menu choices. The user can run any of the operations as explained below.

For the creation of a new record, the procedures creates a temporary facsimile file into which the user enters the new information. Query procedures then check to see that specific rules have been followed.

1. The FEL Order Numbers must be a minimum of 7 digits long, for example, 5527061 is a legal number;

2. The first number, comprising the first 7 digits, must be smaller than the last number, taken from the first and last few digits in the string format, if we take an FEL Order number string, 5527190/301, as an example, the first number would be 5527290 and the last number be 5527301;

3. Any of the order numbers specified must not be any number already in existence in the Forging Status relation.

Any error detected will stop the user proceeding. Finally, when the user has entered the required data satisfactory and proceeds to the creation of the record, the procedure will assign a new ID_No. and the issue letter "A" for it, and automatically enter all the items into a record in the Forging Status relation.

In the record modification operation, the procedure will only load forging definition records used in a related current manufacturing plan. Also the procedure will not allow FEL Order Numbers to be modified because this could result in anomalies in the system and would certainly not be a valid operation.

In record revision operations, the relevant procedure will only load Forging Definition records that have been used in an issued manufacturing plans. The revision operation may involve the decomposition of the original FEL Order Number string into other number groupings. The query rules would check this decomposition has been carried out correctly and does not produce invalid number strings. Once the user has confirmed that a new version of the Forging Definition record should be created, the procedure will create it and modify the records in the Forging Status relation with a new sequential issue letter.

Once the forging definition records have been created, the information can be passed through down to the other planning areas as described in the next few sections.

4.3.3.2 Manufacturing Plan Planning Area

Manufacturing plan planning is normally the second area of data processing in the system, hereafter called area 2. As Figure 4-5 shows, it is linked by the MP_Status relation to the Forging Definition relation, through which, it is further linked to the other planning areas. The MP_Status relation, the Operation and the Test_Units, constitute the component parts of the manufacturing plan. There is a further relation which is not shown in Figure 4-5, RMC_Scrap relation, which stores standard scrap and turnings codes for each RMC (Rationalised Melting Code) recorded in the Manufacturing Plan relation, as shown in the appendix 1-3.

These codes describe different categories of internal scrap and internal turnings depending upon chemical composition. Only certain of these categories can be used for any specific Rationalised Melting Code. The relationships between the RMC codes and the scrap and turnings codes represent metallurgical knowledge built up over many years.

The manufacturing plan data structure is shown in Figure 4-7, with default data to show how records in the different relational tables are drawn together to form one manufacturing plan. Figure 4-8 shows the hierarchy between those different relations.

The data processing operations involved are the creation, modification and revision of manufacturing plans. Although the planning area is at a high level in the manufacturing hierarchy, these operations require information taken from the previous area.

The designing of manufacturing plans is a complex operations. Not only does it specify the operations to be undertaken, but indicates their scheduling as well. Each operation would be allocated to a operation site, with an instruction note and a quality control

Manufacturing Plan

MP_No. 0642 Issue:B

Main form from relation Manuf. Plan

Sub-Form from relation, MP_Status, to indicate all related FEL Order No.(s) and their relevant information such as Customer, Drawing No. and Spec. etc.

Manuf. plan operation details from Operation relaion to record all operations specified in the plan

Test requirements from Test_Unit relation to record all tests and its units requirements

RMC Codes for related information from RMC_Scrap relation RMC:9980/03

MANUF. PLAN

MP_No.	Issue	Name	Date	RMC	Others	QA.Com
0642	B	A. K.	8/1/94	9980/3	texts	BS5750

MP_STATUS

MP_No.	Issue	FEL Order No.	ID Ver.
0642	B	5527065	A
0642	B	5527060/2	B

OPERATION

MP_No.	Issue	SEQ	OP_No	Operation	Comment	QA_act	Prod_Time
0642	B	1	1	Cast Ingot	Text 1	...	W033
0642	B	2	2	Forge	Text 2-1	...	W035
0642	B	3			Text 2-2	...	
0642	B	4	3	Machine	Text 3	...	W036
0642	B	5	4	H_Treat	Text 4	...	W036
0642	B	6	5	U/S Test	Text 5	...	W037
0642	B	7	6	Final M/C	Text 6	...	W038
0642	B	8	7	Test_X	Text 7	...	W039
0642	B	9	8	Doc. Pre	Text 8	...	W039
0642	B	10	9	Dispatch	Text 9	...	W040

TEST_UNIT

MP_No.	Issue	Test No	Test	Test Unit	Comments
0642	B	1	Analy.	Each Forging	
0642	B	2	Stress	Each End	

RMC_SCRAP

RMC	Solid)Code	Turnings	Comment
9980/03	NEL	NEL	Steel G 1
8991/01	ML	CL	Steel G 3

FIG. 4-7 Manufacturing Plan Data Drawn From Different Relations

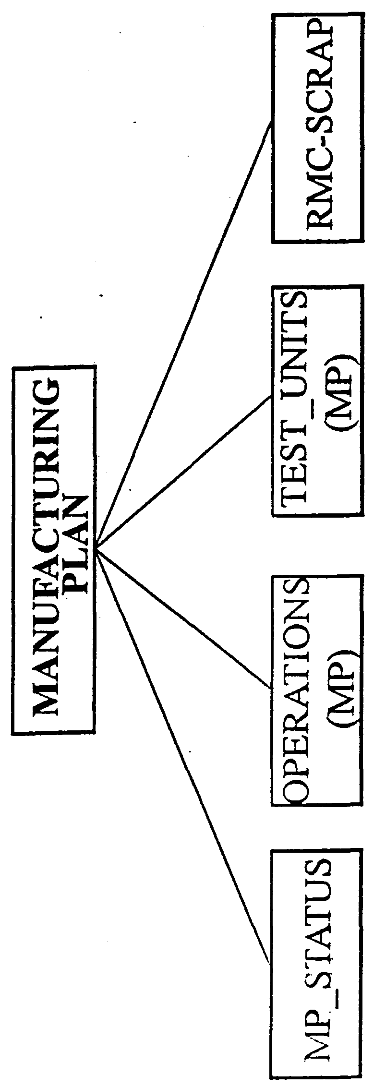


FIG. 4-8 Manufacturing Plan Relations Hierarchy

code. In order to speed up the creation of the plans, the manual system has been modelled by introducing a series of coded master plans.

Each master plan has a unique number as the main key, representing the other composing fields of an RMC code, a heat treatment cycle code and a steel grade. Those fields have been defined as the main relation. Apart from this relation, the master plan relations have exactly the same data structure as the manufacturing plan relations. During daily operation, only the chief metallurgist can alter the master plans according to his experience and the manufacturing results. Once they have been completed satisfactorily, these plans can be drawn on by other planners to design new manufacturing plans for the same or similar forgings. This provides an alternative way to the one described below to create a manufacturing plan.

A. Creating of a New Manufacturing Plan

When a user starts a new manufacturing plan, the system asks for the designation of the FEL Order Numbers to which it would apply. The procedure searches to find a match, or matches, with the Forging Status table. If this search fails, the system informs the user requesting a correction to the order number specified or a check to discover whether corresponding definitions have already been entered previously into the database. Providing the forging definitions do exist, the user can proceed and the manufacturing plan sequence can be started which causes the following events to happen within a few seconds:-

1. A unique MP_No. and the first Issue Letter "A" are assigned for the new manufacturing plan.

2. Since the range of FEL Order number covered by any one manufacturing plan

could correspond to more than one Forging Definition, the procedure decomposes the FEL Order number range entered in the manufacturing plan into sub-ranges to correspond to the Forging Definition records and creates one MP_Status record for each sub-range.

3. The manufacturing plan is created with certain administrative information, such as the current date, the initiator of the plan, etc. A draft plan is then created using the most used set of manufacturing operations, or the manufacturing operations from the coded master plans for the particular specification and forgings. Either of these list of operations appear as a default which can be modified by the user later on.

The operations specify the steelmaking and degassing procedures to be used, the forging and machining procedures and heat treatment required. They also specify the types of testing to be used.

B. Modify Current Manufacturing Plan

Manufacturing plans not yet issued may be modified, since their relevant records are stored separately from these for issued manufacturing plans as facsimile records. The files for manufacturing plans currently under development and issued for use in controlling production have identical data structures. A field in the MP_Status records called 'In_Progress' has the value "NO" for all the current manufacturing plans undergoing modification, A "YES" value in this field indicates that the relevant manufacturing plan is ready to be executed in production. Use of this dual structure is essential since planners normally spend more than a single session in planning, especially when dealing with complex forgings.

Modifications to the manufacturing plan normally involve editing operations which require the operation sequence to be re-arranged. In order to provide some degree of word processing ability within the database, an additional field has been added to records in the operations relation in the manufacturing plan area - the Line Sequence Number (LSN). As long as the line sequence numbers are specified in any sequence of numbers, the procedure rearranges the line and operation sequences so that the operations can be printed in order, however many lines of type are needed to specify each operation.

C. Revision of a Manufacturing Plan

Even though a manufacturing plan has been issued, it may require revision: a customer's requirement may change whilst the forging is being manufactured, or experience with an earlier manufacturing operation may require a later operation to be modified. It is essential that the manufacturing plan issuing procedures conform with the quality assurance procedures within FEL which meet either ASME or BSEN ISO 9002 standards. The data processing procedures must therefore mirror the rules about the manual activities. One example is the evolution of issue letters through the alphabet sequence "A" to "Y" omitting letters "I" and "O".

Once the revision operation is initiated, the programme generates a facsimile set of records from the issued ones and locates them in the files of current records being modified. This allows the user to perform the editing operations described in the previous section.

It will be apparent that there may be two records for the identical FEL Order number string in existence in the MP_Status relation, one with a value of "YES" for the forgings in progress field and one with a value of "NO", the latter being for the plan under modification. The latter can not be connected to any data execution in parts of the database outside the Manufacturing Planning Area, and, moreover, the existence of the revised manufacturing plan in the current manufacturing plan files stops execution of the previous plan.

4.3.3.3 Quality Plan Planning Area

Quality Plans are always created for complex forgings or are required by some customers. In both cases, the quality plan contains detailed descriptions of all the operations necessary to produce the desired forgings. Quality plans are additional to the Manufacturing Plans which normally comprise no more than one page of information.

The quality plan planning area is linked through the QP_Status relation to the Forging Definition in the same way as the manufacturing plan area is linked through the MP_Status. The Quality Plan planning area contains many sub-plans as follows,

1) Quality Plan (QP);
2) Heat Treatment Procedure (HTP);
3) Forging Procedure (FP);
4) Control Document Status List (CDSL);
5) Cleaning, Packing and Protection Procedure (CPP);
6) Mechanical Testing Procedure (MP);
7) Special Procedure (SP).

In Figure 4-5, we have outlined the relationships between the main files associated with the quality plan. Each of these plans has its own sub-relations or related files which are listed in the appendix 1-4.

The two status relations, HTP_Status and FP_Status, state the FEL Orders numbers to which the HTP and FP respectively apply. As shown in appendix 1-4, those two relations have similar field definitions as the QP_Status file. Although all these procedures should apply to the overall total number of FEL orders specified in Quality Plan, it is necessary to have several HTP or FP procedures for one Quality Plan. These two status relations are thus necessary to show the FEL Order Numbers to which a particular HTP or FP procedure applies. The procedure number is the third identifier for both of the procedures.

In all these plan relations, we use a combination of at least two fields to identify a unique record. The first field values are the MP_No. taken from the manufacturing plan, for the same grouping of FEL Order numbers in order to associate them with each other. Before creating any of these plans, the programme will ask the user to enter the FEL Order Numbers to which the plan will apply so that the procedure can read the Forging Status and, from that, read the MP_No. from the top level of the manufacturing plan hierarchy. The quality plan and the manufacturing plan are at the same level in the hierarchy, but, for ease of data manipulation, the manufacturing plan must be created first. This allows the Quality Plan to use the same identifier as the Manufacturing Plan. This identifier is very important since it links all the related plans together so that they may share information. The first identifier is not changed when a plan is revised to the next issue. Once the new issue has been approved, the previous one is withdrawn from the database to archive. The top issues of these plans can therefore correspond with each other as shown in Figure 4-9. Figure 4-10 shows the hierarchy of the Quality Plan and Manufacturing Plan relations linked by the first identifier.

FIG. 4-10(a) shows the manufacturing plan and quality plan hierarchy environment in the form of an ER diagram. Each entity box represents a relation, which has been decomposed into BCNF form. The notation used follows the methods given in [J88].

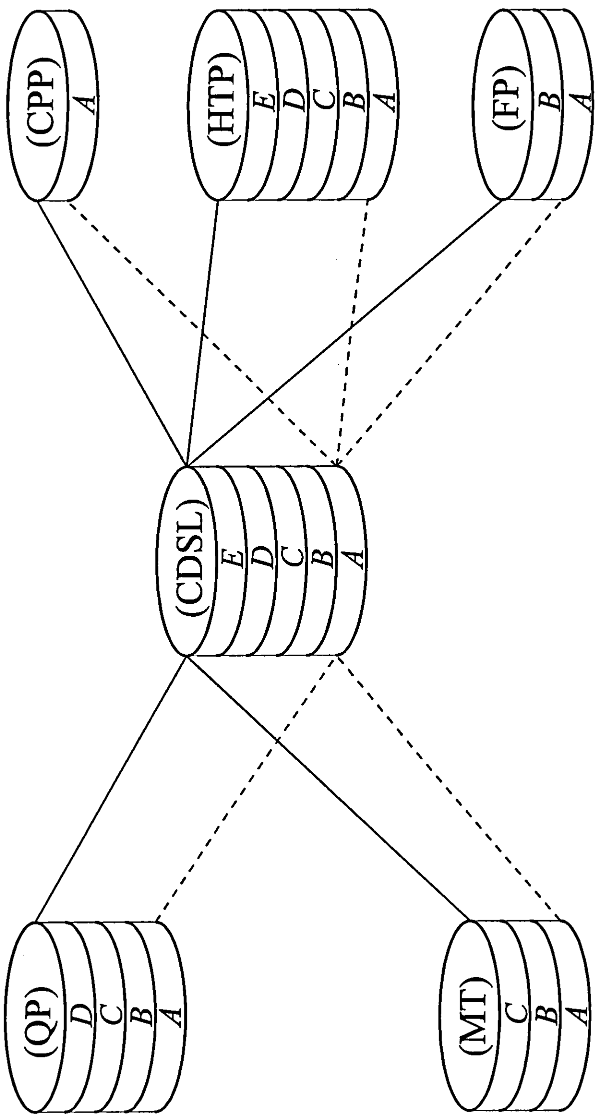


FIG. 4-9 Internal Relationships between Versions for some of Process Plans in Quality Plan Environment

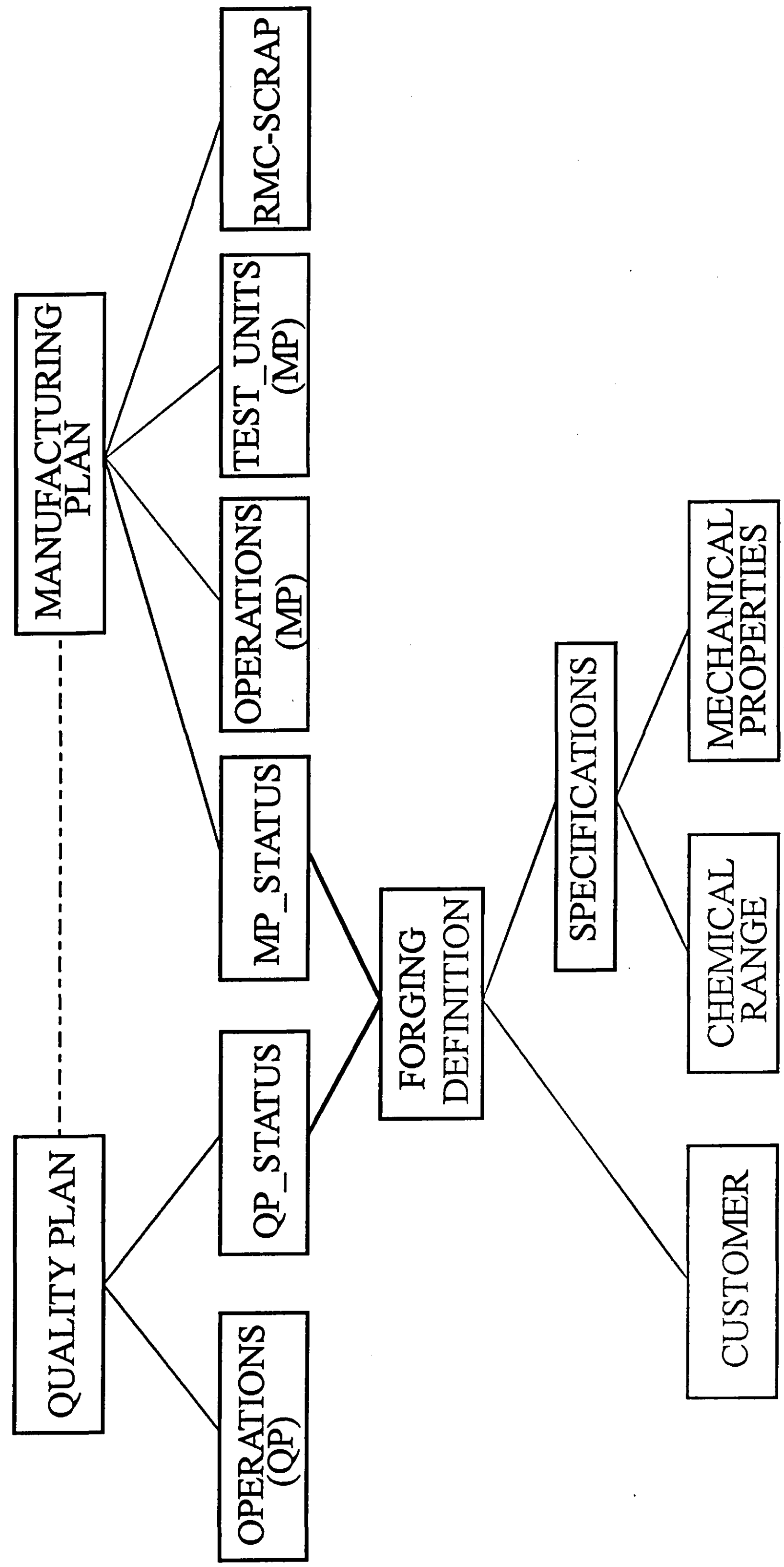


FIG. 4-10 Manufacturing Plan & Quality Plan Relations Hierarchy

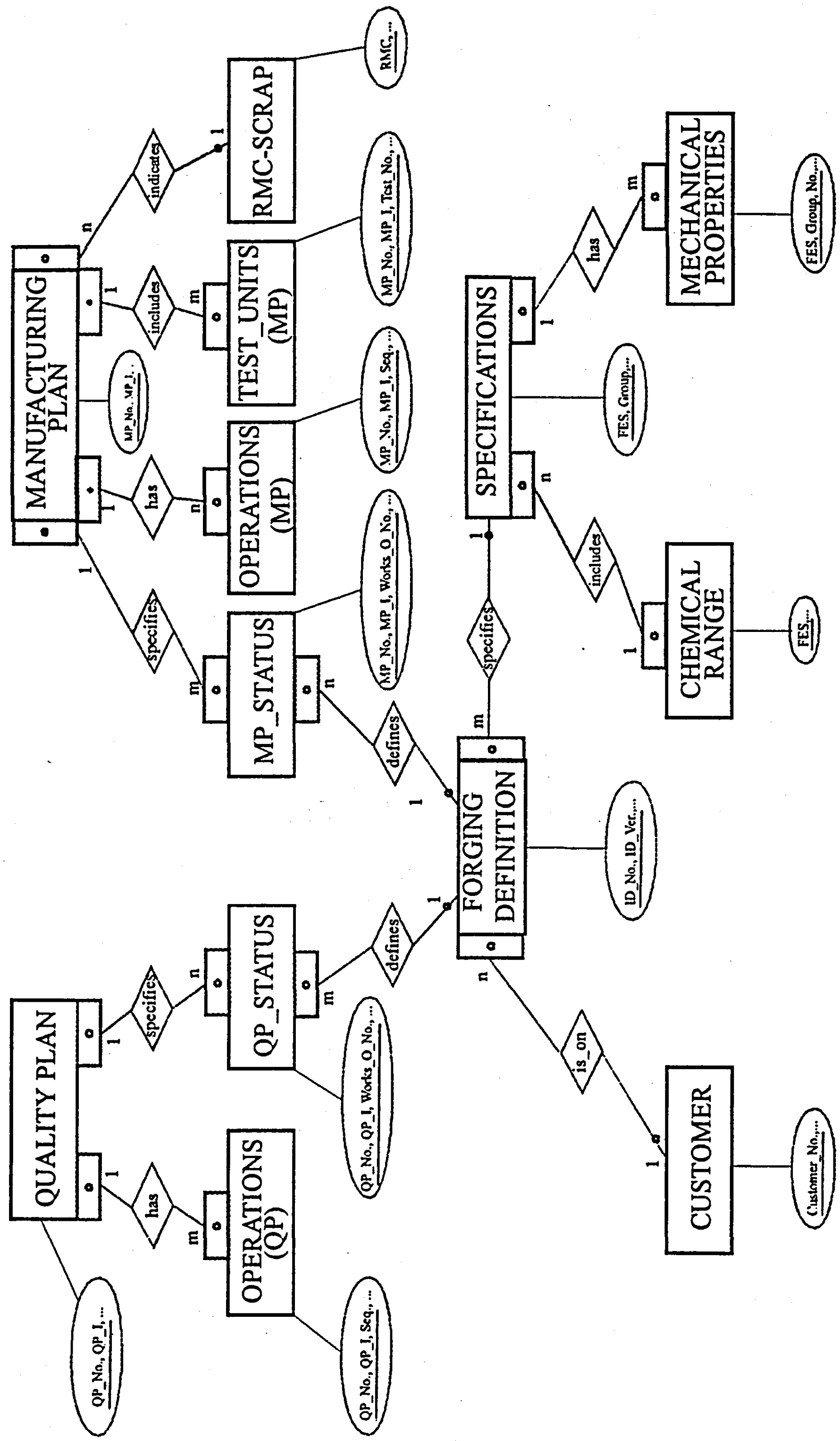


FIG. 4-10a ER Diagram for the Manufacturing Plan & Quality Plan Relations Hierarchy

4.3.3.4 Steel Order Planning Area

Steel Order design is a activity conducted in this planning area, but the area also contains information fed back from the sister company, concerning the steelmaking programme and the cast analysis. It is the most complicated area in the process planning system.

The FEL Melting Range and the Ingot Requirement are the main items to be developed in the design of a Steel Order. The melting range is the chemical composition to be achieved according to customer specifications or requirements, as well as the present capability of the steelmaking processes and cost of these; while the ingot requirement is the designation of ingot sizes and some relevant parameters to be achieved, of which cost factors are also important. Both melting range and ingot requirement are the responsibility of different specialists - metallurgist and forge planner. Metallurgists normally take decisions in this part of process plan designing process. The system must, therefore, be able to pass information across to the forge planners. So that, once the steel order design has been completed by the metallurgists, the forge planners are then able to carry out their part of overall process plan. The system is designed to stop a steel order being issued without the metallurgists having approved the section for which they are responsible.

Once the steel order has been passed to the steel melting shop, the steelmakers are required to present their steelmaking programme to FEL for approval. As soon as the ingots are cast, the cast analyses must be recorded in the database. Both the steelmaking programme and the cast analysis are then checked to see that they meet FEL's requirements.

Appendix 1-5 lists all the relations in this planning area. Among them, FSL_Proc, FSL_Ingot, FSL_Programme constitute the Steelmaking Programme, and Analysis_index and Analysis constitute the Cast Analysis, although neither is shown in Figure 4-5.

1. Steel Order and Ingot Requirements

As shown in Figure 4-5, the SO_Status, Steelord and FEL Melt. Range relations form one part of the steel order, and the relations - Ingot Requirement, Ingot Status and Ingot, form the other. Even if they come together into one process plan, we need to keep two status relations because they serve two different functions. The SO_Status (Steel Order Status) uses a single FEL Order Number format in each record, the Ingot Status uses a string for the group of forgings since a single ingot can be formed into more than one forging.

All these relations except the FEL Melt._Range relation, use the same field combination as the primary key to identify the relevant records. The FEL Melt. Range on the other hand is identified by its unique number, Melt._No., which is recorded in the Steelord relation.

2. Steelmaking Programme and Cast Analysis

Every works order that is required to be cast by the steel melting shop should have its corresponding steelmaking programme and cast analysis reported from the sister company, FSL. It is therefore necessary to add primary keys from both reports into the SO_Status relation for each single order number. Null value would occupy those fields in SO_Status, at the steel order designing stage. These values mean that no relevant report exists in the related files at that early stage of the forging manufacturing sequence. When steelmaking is complete, those two numbers will be assigned to each works order in order to relate all the plans and reports together.

It is necessary to point out that the data manipulations within this area are dealt with in a similar fashion to data manipulation in other areas, i.e. data operations take place on records created in temporary facsimile files before entering those records into the database and issuing them for execution.

Constructing a melting range is one of the major tasks in the steel order design. It requires expert knowledge and experience. The system is designed to assist the experts in tracing a satisfactory melting range quickly and consistently, or to design a new one based on certain constraints. The intention in designing the database is to improve the standardisation of melting ranges and to provide a rational way to modify and extend the melting ranges in use. This will be examined in detail in section 4.6 to show how the system assists in the design of melting ranges.

4.4 Quality Assurance Functions

Quality assurance tasks are considered, at FEL, to be the prime concern that underlines all activities from the start of a contract order, through all the processing steps up to finish and delivery; ensuring that the correct methods and techniques have been used at all stages and that they conform to the required standards. The following tasks have been identified and incorporated into the current database system.

1. Updating specifications;

2. Recording the issue status of process plans;

3. The feeding-back information such as melting programs and ingot analysis, etc.

The selection of these tasks is controlled through the menu system developed as shown in Figure 4-11.

4.4.1 Integration of Quality Assurance Functions

Apart from specifications which already exist in the database to support metallurgical functions, the other QA aspects have not been described. In this section, we first discuss how the QA function is applied to the control of specification withdrawal, and then move our attention to organising and maintaining the issue status of process plans and automating the approval and acceptance mechanisms for melting programmes and ingot analysis.

4.4.1.1 Specification Control

The specifications are actually stored in the current database and entered by metallurgists in the forging definition database planning area (see Fig. 4-5). Their data structure has been described in the section 4.3.3.1 dealing with Forging Definition Planning Area. This section will show how quality assurance functions are incorporated by modifying relations already described.

According to the company's ASME Quality Manual for the control of the steel specifications, previous issues of a specification should be withdrawn once a new version is issued. In a manual system, hard (paper) copies of a specification can be

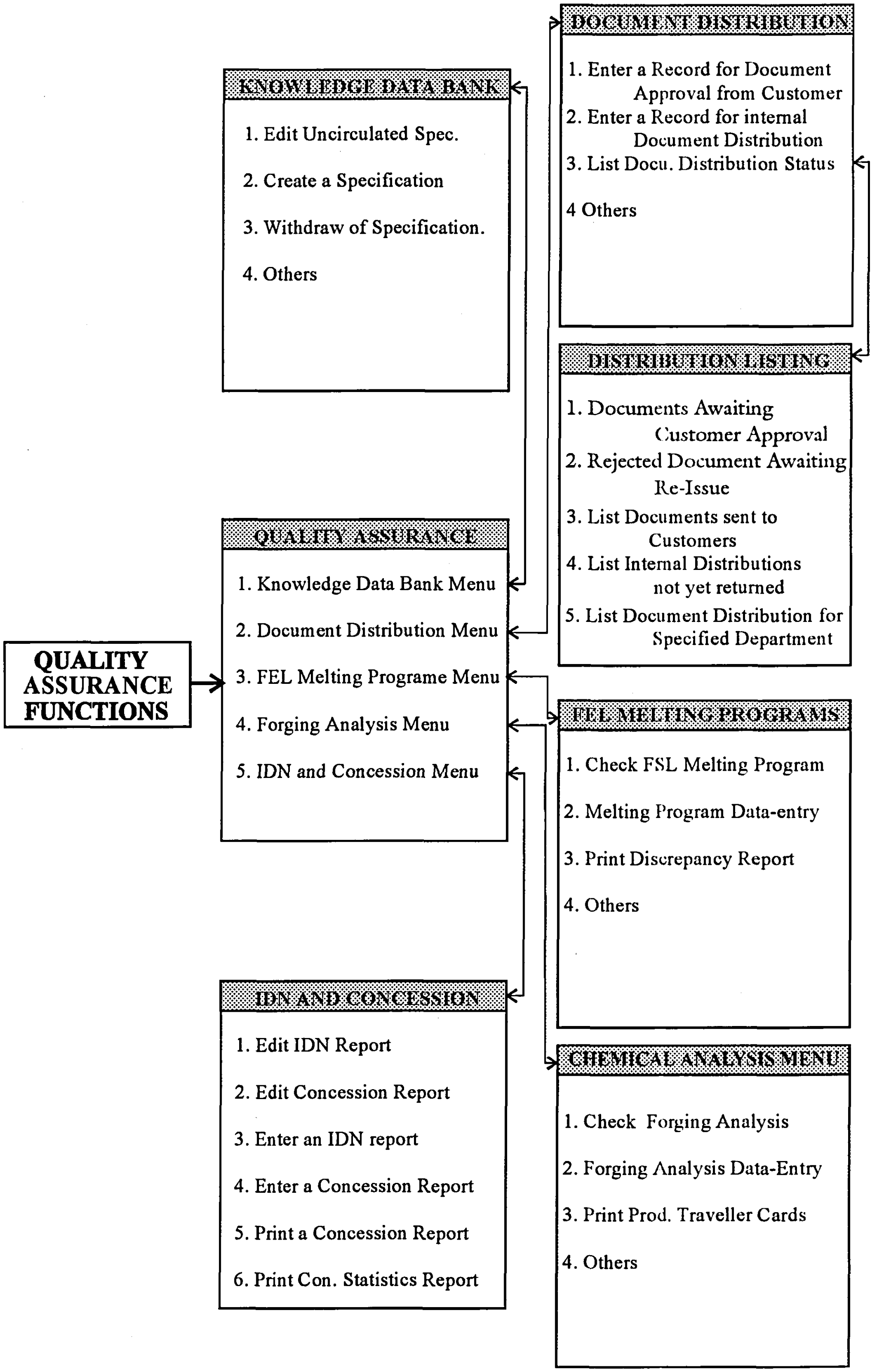


FIG. 4-11 Menu System for Quality Assurance Functions

withdrawn from the specification storage cabinet to be destroyed, while the specification data still remains in place in the process plans that are still to be completed, or have been completed. In a computer based system, one specification record will serve specification used in very many process plans, both in progress and archived: The record cannot be deleted, but must be disabled for use in future process plans, once a updated version has been issued. Therefore,

1. the previous issues of a specification should be disabled for further use but retained in a computer based system;

2. the pointers linking the disabled specification to the records of those process plans that used it should remain in place.

In order to do this, a field, 'ALIVE' was added to the specification relation indicating whether or not a specification is valid. This field is automatically filled with the value "Yes", to indicate a valid issue, as part of the procedure which issues a new or updated specification. Here, the 'ALIVE' field value can have different interpretations if it has a value of "No"; depending on whether the specification has been issued or not.

If a specification has been issued, and the field 'ALIVE' = "No", the specification associated is obsolete and can't be used in any new process planning activity; If the specification has not been issued, the field 'ALIVE' will equal "No", because it has not yet been changed to "Yes" during the issue process, so that the specification record will be a candidate for further modification prior to issue. A further check is necessary, however, before the relevant data entry procedure will allow the specification file to be modified. This check is carried out by examining the process plan records to discover whether the specification has been used in an existing plan. Only if that is not the case, will the procedure allow a record with 'ALIVE' = "No" to be modified within the Forging Definition Planning Area.

In order to maintain the security of specifications in circulation, the field 'ALIVE' is allocated to a security access level higher than that available to the metallurgist: only QA personnel can access the field. This rationalises control procedures within the database system. This function is accessed through the menu item called 'Withdrawal of specifications' amongst the Quality Assurance Functions shown in Fig. 4-11, the query procedure being listed in Appendix 4-2.

4.4.1.2 Process Plan Distribution Status

The current manual system operated by the Company incorporates set procedures for controlling the copies of documents related to process plans and circulated throughout the company. These procedures involve the issuing and withdrawing of documents to and from their points of use. Documents include all process plans in the previous discussed process planning areas. Although a fully dedicated computer system could carry out the corresponding processes automatically, the current system has to be designed to work in parallel with the paper system. Controlling the issue and withdrawal of paper documents must therefore be recognised as specific tasks to be carried out by the computer system, not least because these tasks are specified as steps in the Company's QA procedure which has been prepared to meet ISO, BS and ASME standards.

There are two systems for document distribution status, one for internal distribution and the other for customers distribution. The internal document distribution status system, named Document Distribution Matrix (Appendix 1-7-1), records the issue/withdrawal status of paper documents related to all their internal points of use; while the customer issuing status system, named Document Sent (Appendix 1-7-2), records the same information as the internal system, but also indicates that an approval process is required.

Internal Document Distribution Status

Three relations have been designed for this task as can be seen in appendix 1-7-1 which lists the fields in each relation. The main relation is identified by document name and number. For each record in the main relation, Docu_Distribution, there may be several records in the sub-relations, Issue_Status, and Distr_Matrix.

There is one record in Docu_Distribution for each document (key identifier is document name and number) and 7 fields in each record list the issues of that document that are in circulation. In addition, a further field 'Cleared' can be set to "NO" if there are issues of the document still to be returned. For each issue, including those that have been withdrawn from circulation, one record exists in the Issue_Status relation to record the date of issue and a general purpose comment field. Each record in the Distr_Matrix relation refers to a document and a single point of use (key identifier is document name and number and department name) and lists the issue/withdrawal status, at that point of use, of the each of the seven issues listed in the Docu_Distribution record for the document. The Departments to which the document should be sent can be read from a further relation, Docu_Departments, which contains the standard distribution list for each type of document and which therefore controls the number of records that exist in Docu_Distribution for each document.

The seven fields in the main relation, Docu_Distribution, maintain the correct sequence and order of the document version letters starting from A to whatever, but omitting I, O and Z. Data entry control formula limit the entry possibilities in the first field to "A", "H", "Q", or "X", with other fields progressing through the alphabet. The seven fields in the sub-relation, Distr_Matrix, for each departmental record, indicate whether or not the version of the document defined in the corresponding fields in the main record, has been returned by the indicated value of "Yes" or "No". Because there are only seven fields to maintain the withdrawal status on distributed documents, the maximum number of un-returned previous issues of the document can not exceed 5. This has been enforced by the formula in each field, for example, if a department is holding five issues of a document from "A" to "E", the next issue, "F", of that document cannot be distributed to that department unless the first issue, "A" is returned at this stage.

Limiting the number of fields that can store this type of information to seven saves disc space, but of great importance in respect of the QA function, is the limit of five placed on the number of versions of a Specification, Process Plan, or any other printed document; that can be in circulation within the Company at any given time.

Customer Issuing Status

The maintenance of the customer issuing status is rather simple compared with the internal document distribution status, but shows same equally important aspects. Two relations, Docu_Sent and Sent_Status, have been designed for this particular task as can be seen in the appendix 1-7-2. Document name and number are the key fields identifying each record in the main relation, Docu_Sent. For each record in the main relation, there may be several records in the sub-relation, each referring to a different issues of the document that the customer is holding and can containing the field, 'Status' which can be set at 'Awaiting re-issue', 'Approved' or 'Not-Returned'.

4.4.2 Automation of QA Functions

It has been observed that QA control activities are enormous and time consuming. In a manual system of QA control, QA personnel need to fill in tables for recording the distribution of each document for each department and customer. For hundreds of orders with up to ten process plans, there are thousands of tables to be filled and constantly checked. For a big contract or complex forgings, it is quite often that a process plan version development can be up to more than 10 issues. This makes the situation much more difficult. The need for a rationalised computer system is obvious, and benefits that such a system can bring are significant. We have already shown how the QA function can be incorporated in the system in relation to the control of specifications and other documents. Other functions such as diagnosis checking will be discussed later in this chapter, but below automation in other areas is described:-

4.4.2.1 Internal Document Distribution Check

Two automation facilities have been provided, one is to simplify the distribution of issue "A" to departments, the other is for listing departments which are holding documents due for return. They are run under the menu of Document Distribution in the Quality Assurance Functions (Fig. 4-11).

<u>1. First Issue of a Document Distribution</u>

A control query procedure is designed for entering the records of internal document distribution as can be seen in appendix 4-2-2. Each time, when the first issue of a document is issued, the control procedure will create a record in the relation, Docu_Distribution, and create another record in the sub-relation, Issue_Status, by putting two more values such as issue letter "A" and issue date. It then reads the distribution requirement from the Docu_Department file and copies them into document distribution relation file, Distr_Matrix in addition to the issued document name and number. As a result, the rule stored in the first field among the seven 'Issue' fields in the main relation, Docu_Distribution, will be triggered to generate "A" and the corresponding first field in the sub-relation, Distr_Matrix, is set to "No" to indicate that the issue sent to a department is not to be withdrawn.

2. Listing of a document distribution

A query procedure has been designed to list departments which are holding un-returned documents, see appendix 4-2-3. In order to speed up this process, the field, 'Cleared', in Docu_Distribution is first examined. If 'Cleared' equals "Yes", no un-returned documents remain in circulation, so that the query procedure needs only examine those records with 'Cleared' equal "No". For these records, the procedure then examines the corresponding records in the relation, Distr_Matrix, to find out which departments hold the un-returned issues. The final results can be sent to screen, or printed on paper for sending to the relevant departments. This system has greatly improved the speed and reliability with which un-returned documents can be traced within the Company.

4.4.2.2 Customer Document Status

The procedure for reviewing the status of documents sent to customers is similar to the internal procedure, but simpler. A field called 'Finish' in the relation, Docu_Sent, can have the values "Yes" or "No" to show whether or not the document sent to the customer has been cleared in the reviewing process.

Separate query procedures have been designed for different functions and all of them are run under the menu of Distribution Listing in the Quality Assurance Functions, see Fig 4-11. The basic logic of these procedures is that they examine the records in the main relation, Docu_Sent, with 'Finish' = "No", and check the sub-relation, Sent_Status, to list documents awaiting re-issue, approved for issue or not yet returned:-

1. Awaiting re-issue: listing all issues of documents which need to be re-issued;

2. Approved: listing all issues of documents which have been approved by the customer;

3. Not-Returned: listing all issues of documents which should be returned.

This facility improves working efficiency between the QA department in terms of reducing waiting time and speeding up the processes by which documents are sent to and withdrawn from FEL's customers.

4.5 Test Results Functions

The NDT and the mechanical tests to be carried out on each forging are specified in the Manufacturing Plan to meet standards set in the forging specification. The FEL Metallurgical CAPP system allows the results of these tests to be recorded for each forging and related to the corresponding forging definition file. The NDT test are carried out within FEL on the shop floor, but the mechanical tests were carried out within another sister Company.

The FEL Metallurgical CAPP system, at its current stage of development, allows data related to these test to be stored and used as the source for printed test documents, but does not carry out any approval function. Figure 4-12 has shown a menu system developed for the test result functions.

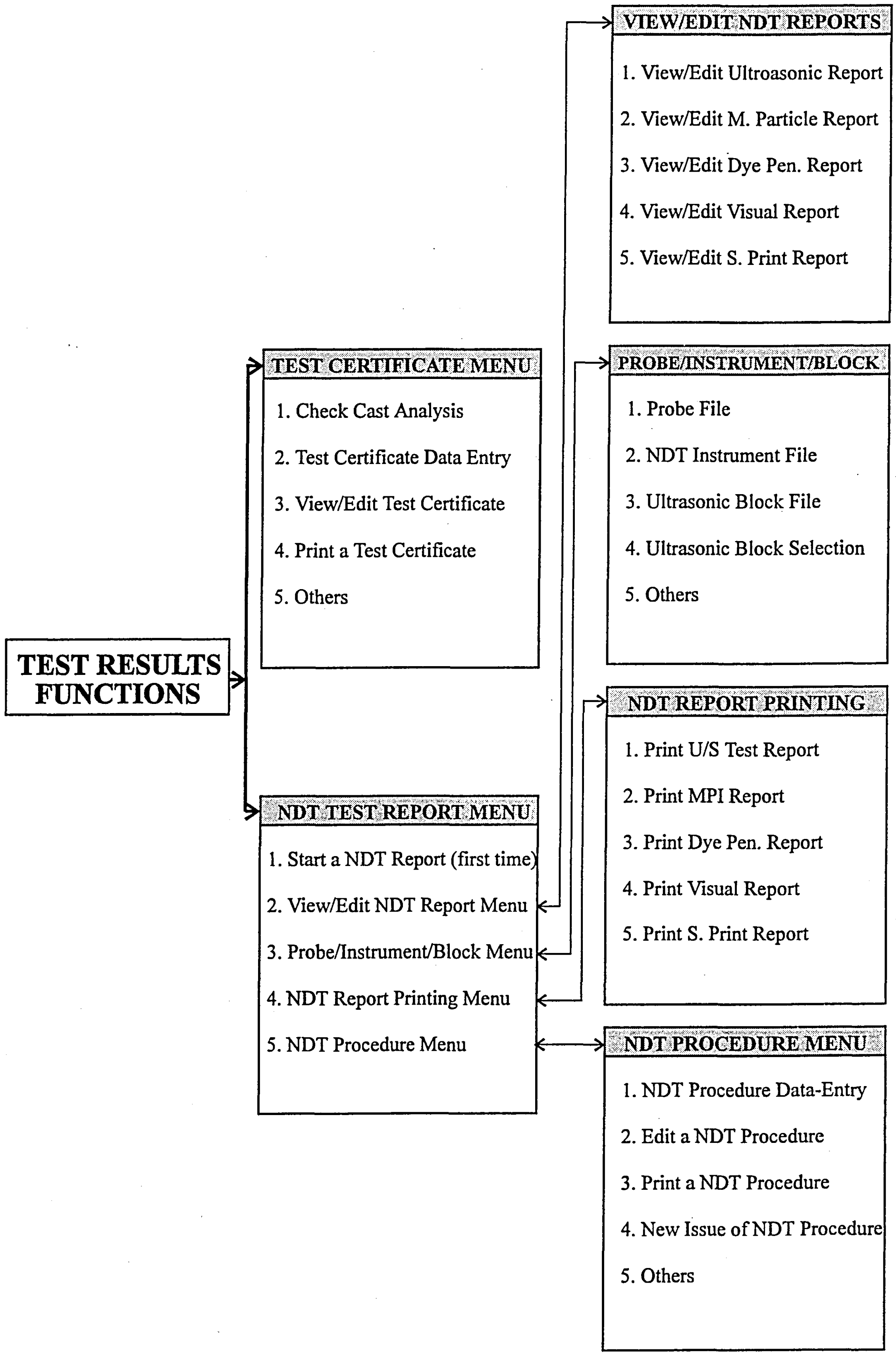


FIG. 4-12 Menu System for Test Results Functions

For each mechanical test, the data recorded include the test conditions, such as location and direction of the test, the test certificate number and the numerical values of the test results. For the NDT test, such as dye penetrant, sulphur print, ultrasonic test, etc., the data recorded includes the test conditions, the reference of any visual test record, and the verdict of the test operator. One relation exists for each type of NDT test and two relations for all the mechanical tests with pointers linking records to the relevant forging definition file, as shown in Figure 4-5. The relations involved and the fields that they contain are shown in Appendix 1.6.3.

Checks built in to the data entry procedure for the test records ensure that records are only entered where the Manufacturing Plan specifies that the test is to be carried out, and that test data is not entered twice. The system was designed so that the NDT test records could be created by the test operators on the shop floor, although this facility was not implemented. Procedures were also developed for modifying test records and for QA personnel to print test result documents to send to customers (see appendix 4-3). These procedures ensured that, where test results have been incorporated into a test result document, the corresponding test records cannot be altered. A further procedure allows QA personnel to examine the test records to ascertain what results are ready to be sent to customers.

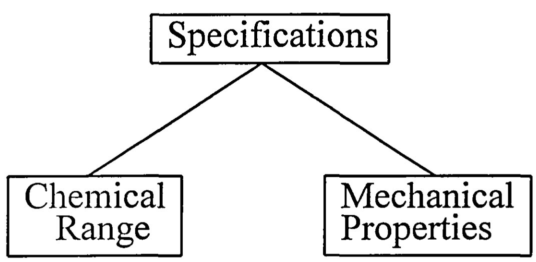
Although the test records could be used to indicate the success or failure of past manufacturing plans before they are used as the basis for new plans, a procedure for making the necessary checks was not developed.

4.6 Intelligent Database System Approach

Until now, the description of the CAPP system has concentrated on the handling of data concerning Manufacturing and Quality Plans, Specifications, etc. and on the procedures necessary to keep that data secure within a multi-user Process Planning Environment. Although these procedures have utilised the logic facilities offered by the database's query language, the use of these facilities to assist the Metallurgist's planning process has not been discussed. This section will describe the assistance that has been provided in three areas - the linking of the mechanical properties and chemical range aspects of Steel Specifications, the designing of steel melting ranges and the approval of steel making procedures and cast metal analyses. Since these aspects of the CAPP system offer assistance to the Metallurgists drawn from past manufacturing experience within FEL, they are considered to be intelligent aspects of the CAPP system, especially since they allow the base of past experience to grow and accumulate as the system is used.

4.6.1 Specification Manipulation

There are three relations in the specification database, as shown in the following diagram:-



The reason for these three is that any single chemical range (a range of chemical compositions for up to 22 elements) can give different ranges of mechanical properties due to the specific internal structure produced by the heat treatment cycle selected, and these properties can vary with position and orientation for certain but not all of the forgings manufactured.

A typical tupple in the Specification relation is shown below for which the key is 'FES' and 'Group' and therefore defines a certain Chemical Range (FES is the key for the Chemical Range tupple) and a specific set of heat treatment cycles. This set of cycles results in specific sets of Mechanical Property ranges, varying, depending on ingot type, with position and orientation. A one to many relationship therefore exists between tupples in the Specification and Mechanical Properties relations.

FES	Group	Cond._1	Cond._2	... ...

Legend:

FES - FEL Specifications,
Group -a group of mechanical properties,
Cond._1 - Condition 1 for heat treatment cycles or forging size,
Cond._2.- Condition 2 for heat treatment cycles or forging size.

It is obvious that considerable manufacturing knowledge is encapsulated in this system, describing how specific sets of mechanical properties can be achieved at different positions and at different orientations in steel forgings of defined chemical composition. This knowledge is based on experience within FEL and, in some cases, within FEL's customers. The relationship between tupples in the Mechanical Properties and tupples in the Specification relation can grow dynamically as that experience is extended.

The FEL CAPP system allows the knowledge that the experience has generated to be available to the Metallurgist during the design of the manufacturing and quality plans, so that the Specification Database assists Metallurgists in their process planning tasks.

4.6.2 Steel Melting Range Design

This section of the FEL CAPP system allows the Metallurgist to design a steel order and a melting range (maximum and minimum chemical compositions of up to 22 elements) for a forging to match the Chemical Range of the Specification for that forging. At first glance it might appear that no design decisions are involved since the Chemical Range similarly defines maximum and minimum chemical compositions for those elements. However, accumulated manufacturing knowledge within FEL allows a considerably more effective procedure to be adopted. This knowledge concerns the interactions between factors such as solute elements levels in steel, heat treatment cycles, required mechanical properties, raw material availability and costs, and steelmaking procedures and costs, etc. In the light of this knowledge, the metallurgist can devise a melting range that, whilst lying within the Chemical Range of the Specification, will optimise the cost effectiveness of the steelmaking process in achieving the final desired properties in the forging.

The metallurgist's task in designing a steel order consists of designing the steel melting range, and defining certain other aspects of the steel procedure, such as deoxidisation practice, analysis requirements, etc. In the FEL CAPP system, the melting range and the other steel making requirements are stored in separate relations, FEL_Melt.Range and Steelord. The FEL_Melt.Range has a structure similar to that of the Chemical Range relation in the specification database, apart from that the key identifier which is the Melting_No. Each Melting_No. identifies a unique tupple maximum and minimum compositions for 18 pre-defined elements and for 4 other elements defined by the users, the melting range designer and the date that the melting range is approved. The Steelord relation stores the basic steel making requirements together with the Melting_No., its key identifier being two key fields, ST_No. and Version. The 'Steelord' also records the MP_No. and Issue (key identifiers for the Manufacturing Plan) to link these two plans together, 'Approve' to indicate whether or not the steel order has been approved by the metallurgist. The tupple also records the date of that approval.

The parallel relation 'Ingot_Requirement' has the same key as 'Steelord' and is used by the ingot designer, after the metallurgical aspects of the steel order have been approved. The ingot designer determines the number, shapes and sizes of the ingots into which the steel furnace charge is to be cast, sometimes combining several forgings into one ingot or combining more than one furnace charge into a single ingot. This is another area where experiential manufacturing knowledge is involved, although it is regarded as outside the metallurgical function within the FEL and has not, therefore, been incorporated into the database. However, the decision made by the ingot designers must be recorded, and the relations 'Ingot' and 'Ingot_Status' record how each original FEL order string is decomposed into separate forging numbers and recombined into individual ingots, each with its own identifier.

Figure 4-13 shows a diagram of the metallurgical function in this data processing sequence. When the process begins, information must be entered in order to allow the programme to search through historical data to locate possible candidate melting ranges from those used previously. The search and query processes are described below with explanations of each stage.

As described previously, there are four relations, Forging Definition, Specifications, Steel Order and Melting Range, involved in this data processing. As can be seen in Figure 4-5, they are in the forging definition planning area and the steel order planning area. From the given forging identification number, i.e. the FEL Order Number which identifies each item to be produced, the specification ID is generated from the Forging Definition relation. This allows the previous steel orders that have the same specifications to be selected from the relation of steel orders that have been issued. This

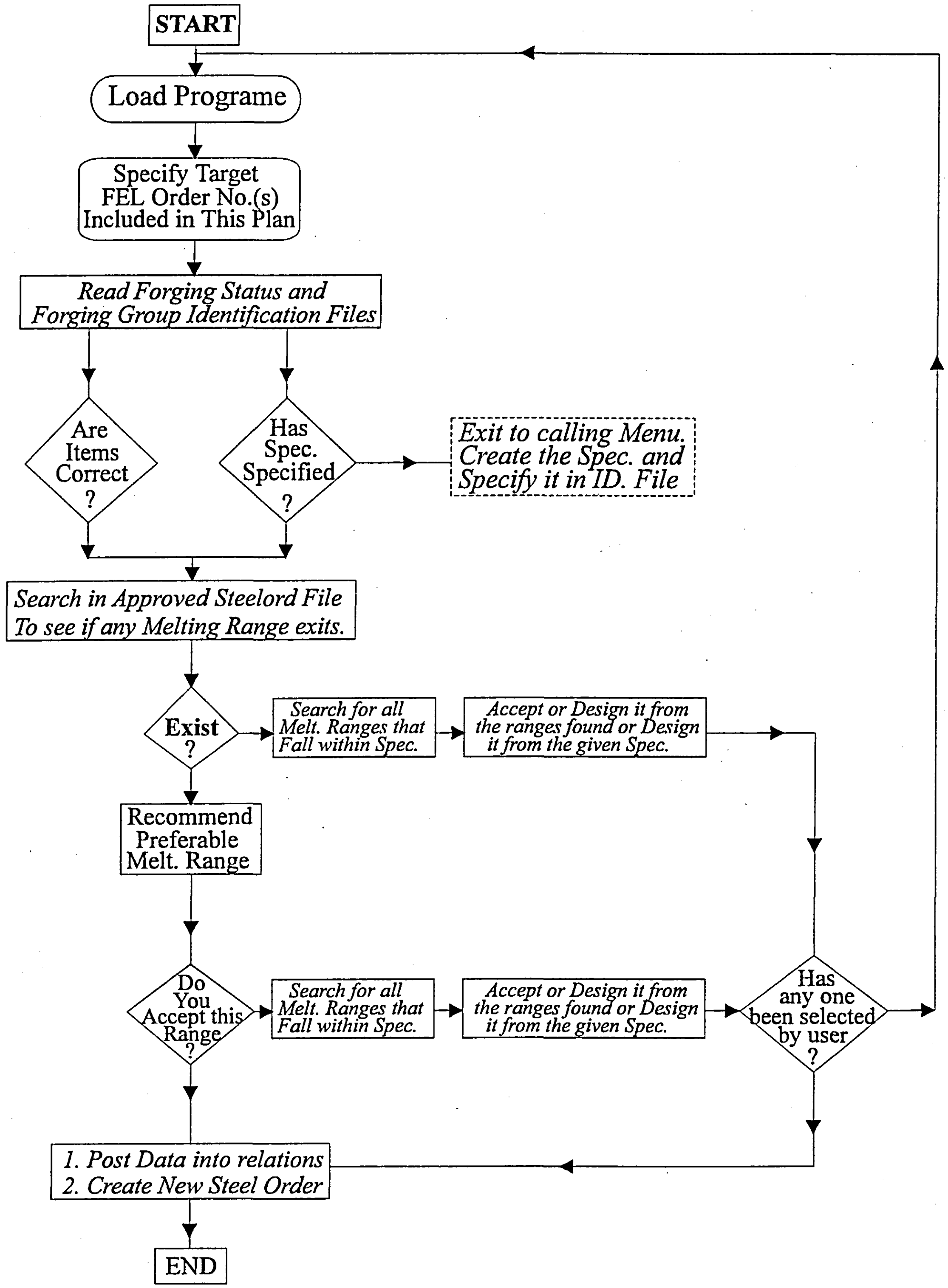


FIG. 4-13 Process Planning Logic for FEL Melting Range Design

restriction is very important because only issued steel orders can be considered as part of the shared metallurgical knowledge. Finally, corresponding steel melting ranges can be retrieved, but once again, only those that have the highest value of the melting range key - the highest key value representing the most recent design for a given specification.

Rule 1: Always select previous steel orders for which the specification is recorded in the current forging definition record, provided that the steel orders have been approved.

When Rule 1 is evaluated successfully, this triggers another rule, Rule 2, to find highest value of the melting range key.

Rule 2: Always retrieve the highest value of the Melting Range key among the candidates.

Provided neither Rule 1 nor Rule 2 has failed this procedure will display the melting range most recently used for the current specification for the metallurgist to examine. Rule 1 & Rule 2, are the generic rules that are modelled in this query procedure. Through the evaluation of these rules, the metallurgist's knowledge stored in the database can be presented to the current user within the steel order planning area.

Up to this point, the main query is about to finish. According to the metallurgist's response, one or the other of subsequent actions will be initiated.

Action 1: If the recommended FEL Melting Range has been accepted, a new steel order record is created and then the metallurgist can enter the other required data into the 'Steelord'.

If this occurs, we can say that the system has automatically reached a solution. What the system has done is to model the human process of searching through existing knowledge. The metallurgist can, however, still design on original melting range.

Action 2: If the recommended FEL Melting Range has not been accepted, a sub-query would be activated to load all the previous melting ranges designed for this specification so that the user can select one of them or design a new melting range by modifying one of them.

This process is necessary. It provides an alternative way of presenting and accumulating metallurgical knowledge.

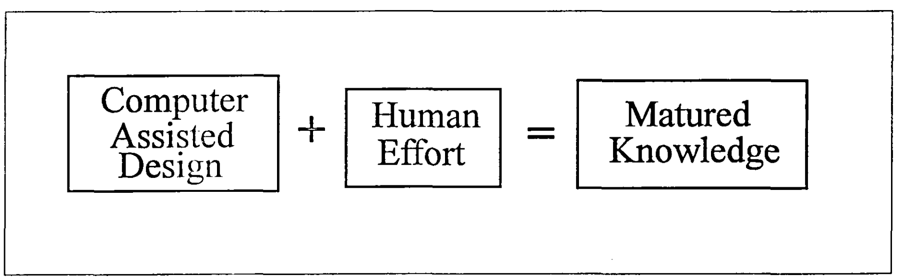
Action 3: If the specification is new, the system has no relevant knowledge. This requires the system to carry out an imprecise search in which it seeks existing melting ranges that approximately match the new specification. In order to minimise data processing time the first six element fields in the FEL Melting Range data file are indexed. The matching search only involves the top 10 elements and is only carried out amongst melting ranges that have been used in approved plans.

Once a melting range has been selected satisfying this match, the system provides a communication interface between the machine and user in order to assist the user to evaluate the selected range. It will inform the user why the selection has been made, who designed the selected range and when it was designed, and with which specifications it has previously been used. Based on these and some other constraints, it is relatively easy for the user to choose or reject a selected melting range. Once a choice has been made, the user may accept or modify the chosen range.

In all the above cases, the constraints applied ensure that no duplicated melting range can be stored and that the melting range created must be within the specification range. Within these constraints, the user has the flexibility to select an alternative solution including modification of the range to better suit the current requirements.

These techniques ensure that the knowledge base data files are maintained at the minimum storage requirement for the available knowledge and that duplicated records are avoided. It uses the overlap structure between specifications and melting ranges as shown in Figure 4-14. The Melt. 1 (Melting range 1), originally designed for Spec. 1 (Specification 1), can be used for Spec. 2; And Spec. 3 has 3 melting ranges, of which the Melt. 3 is developed from Melt. 2, and the Melt. 2 from Melt. 1.

This system allows metallurgical knowledge to be accumulated continuously - the new knowledge matures. Before any steel order is issued, it can be modified in the light of new ideas and information. We can represent this knowledge development using the following formula:-



Remembering and recalling past performance plays an essential role in the steel melting range design process. The approach adopted here is similar to case-based planning [HS91], but offers a better procedure for drawing on past experience in designing new cases.

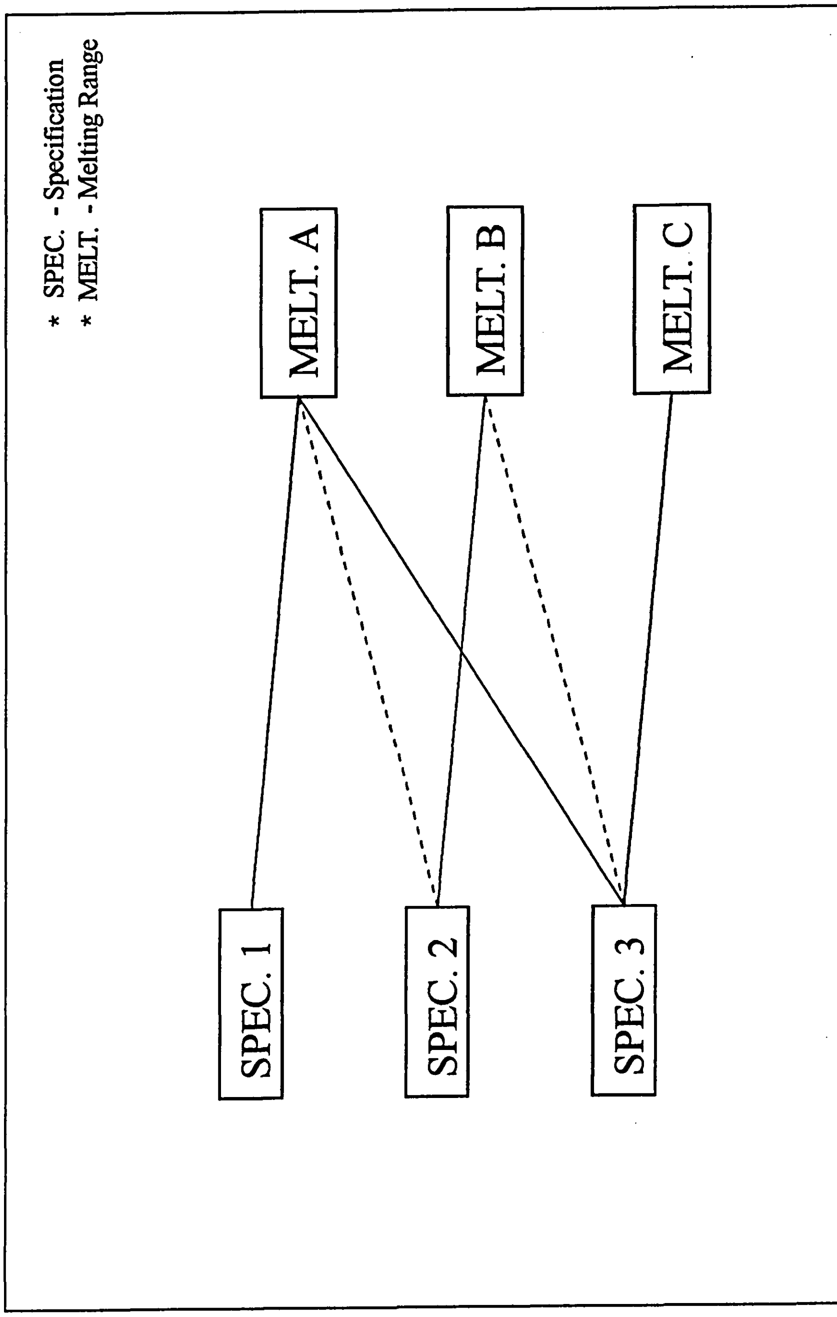


FIG. 4-14 Overlap Storage Structure between Specifications & Melting Ranges

4.6.3 Approval of Steel Making Procedures and Cast Analyses

Two reports from the sister Steelmaking Company require checking and approval - the steelmaking programme and the cast analysis. At first glance, it would appear that these two checking procedures are of a different nature. Approval of the cast analysis involves an automatic check on each value in each chemical analysis field to determine whether it lies between the maximum and minimum values specified in corresponding fields in the specified FEL_Melt.Range tupple. Routine checking of this type is done more reliable by computer than by human experts.

Approval of the Steelmaking Programme, on the other hand, involves checking the steelmaking steps to be used by the Steelmaker against manufacturing rules established within FEL. As will be seen in section 4.6.3.2, however, the use of 'rule templates' allows this procedure also to be carried out by the computer automatically.

4.6.3.1 Approval of the Cast Analysis

The FEL melting range specifies the maximum and minimum values of the compositions of up to 20 elements. We, therefore, have two equal arrays of values:-

$$MA = \{X_1,\ X_2,\ X_3,\ X_4, \ldots X_{22}\}, \text{ and}$$

$$MI = \{x_1,\ x_2,\ x_3,\ x_4,\ \ldots x_{22}\}.$$

Where X_1, X_2, X_3, X_4, ... X_{22} and x_1, x_2, x_3, x_4, ... x_{22} are the maximum and minimum values of the compositions of elements.

The cast analysis is a further array of equal size:-

$$CA = \{\chi_1, \chi_2, \chi_3, \chi_4, \dots \chi_{22}\}$$

It is obvious that the cast analysis values must fall within the designed FEL melting range, so we have the following equation:-

$$[x]_i < [\chi]_i < [X]_i \qquad \text{for } i = 1 \text{ to } 22$$

The FEL_Melt.Range and the Cast Analysis relations list the specified and user defined elements in the same order, so it is easier to design a query procedure to examine the validity of the above equation. Selection of the records to compare is achieved through the central relation, Forging Status, in the database logical structure presented in the section 4.3.1 and shown in Figure 4-2.

The comparison process is simple and straight forward. If the result shows any difference that is confirmed not to be a typing error, a procedure can be invoked to produce a discrepancy report showing the differences as the FEL's QA procedure requires.

4.6.3.2 Approval of Steel Making Procedures

Where ingots are over a specific diameter, FEL's QA procedures specify that the steelmaking procedure used to produce the ingot must be approved, before the ingot can be accepted, against manufacturing rules established by FEL's Technical Director. This approval procedure was incorporated into the FEL CAPP system by transcribing these manufacturing rules into rule template tupples, and seeking a match between the actual procedure used and one of these tupples.

The manufacturing rules governing steel teeming procedures and slag and degassing procedures are of the form:-

IF {conditions are true}, THEN {operations} should be ...

Typical example could be as shown below,

1. Rules governing teeming procedures:-

A typical rule:-

IF the ingot >= set diameter and
it is [Al] killed and
it is not [1 CrMoV] steel
...

THEN
the nozzle size should be X_1 mm;
the aim teeming temperature should be X_2 °C, over the steel liquidus;
A new spray limiter should be applied;
the ingot head should be filled to Notch minimum;
and the liquid spare metal weight should be X_3 Tonnes.

with a further independent rule,

2 Rules governing slag and degassing procedures:-

A typical rule:-

IF the steel is VCD treated and

the ingot weight is between a specific range in tonnes,

THEN

the processes should be,

1) Single slag VAD/LF degas and
2) Double slag furnace .

These rules follow the typical syntax of Expert system rules, but they are not expressed in the FEL CAPP system as logical rules. Instead, rule templates are used to draw {VARIABLES} from the database. Therefore, these rules become,

1. Typical teeming procedure rule:-

IF Is the ingot equal and larger than a specific diameter, {YES}, and

is the steel [Al] killed, {NO}, and

is the steel [1 CrMoV] type, {YES},

...

THEN

the nozzle size should be {X_1} mm;

the aim teeming temperature should be {X_2} °C over the steel liquidus;

a new spray limiter should be applied, {YES};

the ingot head should be filled to {NOTCH} minimum;

the liquid spare metal weight should be {X_3} Tonnes.

2. Typical slag and degassing procedure rules:-

IF the steel is {VCD} treated and

the ingot weight is between a specific range in tones,

THEN

the processes should be

1) {2 Single slag VAD/LF degas} and

2) {Double slag furnace}.

Where the brackets { } enclose the variable value that is stored in the rule template. These values are stored in tupples in two relations, as shown in table 1 & 2 below.

Table 1. Teeming Procedure Rule Template Relation

No.	Igt_X	CrMoV	[Al]	Spare	Nozzle	Temp.	Limiter	Head_To
01	YES	NO	YES	X_{3-1}	X_{1-1}	X_{2-1}	YES	NOTCH
02	YES	YES	NO	X_{3-2}	X_{1-2}	X_{2-2}	YES	NOTCH
03	YES	YES	YES	X_{3-3}	X_{1-3}	X_{2-3}	YES	NOTCH

Table 2. Slag & Degassing Rule Template Relation

VCD	Wt.1	Wt.2	Process 1	Process 2
YES	X1a	$X1_b$	DOUBLE SLAG FURNACE	2 SINGLE SLAG VAD/LF DEGAS
YES	X2a	$X2_b$	SINGLE SLAG FURNACE	SINGLE SLAG VAD
YES	X3a	$X3_b$	DOUBLE SLAG FURNACE	SINGLE SLAG VAD

Legend:
Igt_X - a specific Ingot size;
[Al] - Aluminium;
Spare - Spare Liquid Steel;
Nozzle - Casting Nozzle size;
Temp. - Casting Temperature;
Limiter - A Cast Protection Tube;
Head_To - Cast Level in Ingots;
VCD - Vacuum Carbon Degassing;
Wt.1 & 2 - Ingot Weights
Process 1 & 2 - Steel making processes.

When the program is running, it first reads the steel order data so as to determine FEL's requirements, and then uses these to select the relevant rule tupples from the relations described above. This selection process is carried out by matching the steel order requirements with the values in relevant fields in the rule tupples. The rule tupples selected in this way specify the approved steelmaking procedure and this can then be matched with the procedure actually used by a simple field comparison process. Should this matching process fail, relevant data is presented in screen tabular form to the user who can then make a decision about the issue of a discrepancy report.

This routine is quite important in the way it enhances the work of the metallurgical experts in evaluating the steel making programmes. The manufacturing rules can be easily modified or expanded in the light of experience within the rule base and without the need to change the system programme, through use of standard data entry procedures. This method of encoding process knowledge is highly appropriate within a closely controlled quality assurance environment. Although new rules can be added as steelmaking processes are modified, the structure of the interrelated process decisions is carefully prescribes in a manner that maintains the required quality control manufacturing discipline.

4.7 Summary

This chapter has described the FEL CAPP database system for multiple functions in the application domain, the metallurgical function for process plan planning, the QA function for the overall control of the quality assurance and the test results function for the automation of document printing process. Through the presentations of three typical examples in section 4.6, the complexity of the engineering knowledge is revealed. With the complex database structure based on the core of engineering logic, relations and relationships as the core, the knowledge and the appropriate mechanism to gain new knowledge have been established so that the system can:-

* improve the efficiency of the planning process: with a growing case list of completed tasks and solutions maintained in the system, which supports case-base process planning, rather than the generation of process plans from scratch.

* improve the quality of planning results: human expertise is involved in this planning process. Interaction between the user and the system optimises the results by enhancing the individual's ability to remember successful past plans. It also provides a 'corporate memory' for the group of process planners, allowing them to build on one another's past experience.

CHAPTER 5

DATABASE ENHANCEMENT

5.1 Introduction

This chapter investigates the enhancement of the FEL CAPP database system in two ways: using an expert system and using a high level language programme both as external functions. Based on the relational database structure discussed in the previous chapter, the enhancement aims at manipulating the system rules and data for the process planning. The example chosen for testing the connection of the database system with an external expert system is the approval of the FSL steelmaking procedures. The example chosen for testing the linking of an external programme to the database is the performance of certain complex scientific calculations needed to support process planning decisions. The important common factor is the exploration of external enhancements to the database system approach to process planning.

5.2 The Coupling of the Database with an Expert System

In chapter 2, the methods of coupling an Expert System with a Database System have been reviewed. Unlike the bulk of such coupled systems, which are based on the expert system, the system developed here is based on the database system: the expert system being treated as an external component and called as required. The CRYSTAL [C87] expert system shell was chosen for this work because it has been used for engineering application with some success [BG89][RB94].

5.2.1 System Architecture

An architectural overview of the experimental system is shown in Figure 5-1. The system is built by linking a query sub-system and control routine, within the database

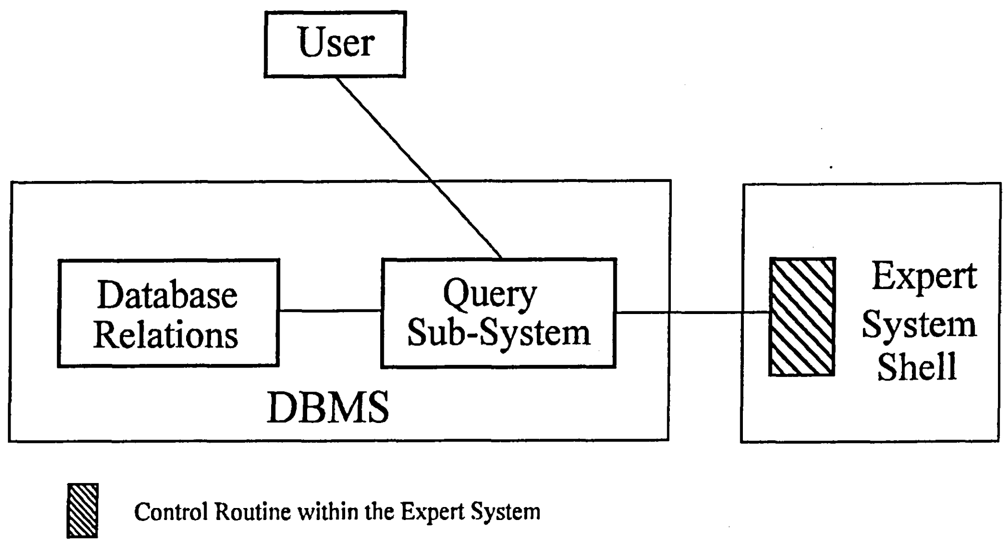


FIG. 5-1 Database System Enhanced with an Expert System

system, to the expert system.

This system has two main constituents:- a query sub-system within the FEL CAPP database system and a control routine within the Expert System. The query sub-system controls the overall operations and the control routine within the expert system provides a route for entering information into the expert system for evaluation.

The small expert system that was developed incorporated the FEL manufacturing rules governing teeming procedures and the slag and degassing procedures. The manufacturing rules governing teeming procedures, for examples, were represented by 18 CRYSTAL rules as compared with 6 rule template tupples within the database representation (see section 4.6.3). The expert system, as a whole, consisted of some 100 rules and 184 commands (see appendix 5 for a list of the rules). These rules were built into CRYSTAL in the form of rule conclusions and rule conditions. Conditions, commands and keywords are used to allow CRYSTAL to evaluate whether a rule is True or False.

Rule Conclusions:-

In CRYSTAL, a rule conclusion is made up of a sentence, consisting of up to 50 characters without any required syntax at all, which can fail or succeed. For example:

FSL steelmaking procedure is OK.

This is what the knowledge base in CRYSTAL will be trying to prove.

Rule Conditions:-

A condition is a piece of information which CRYSTAL will use in order to try and prove a rule. For example:

> FSL steelmaking procedure is OK,
> IF all parameters conform to FEL requirements.

The condition is: If all parameters conform to FEL requirements.

Obviously, if all parameters meet FEL requirements, then the rule succeeds, i.e. FSL steelmaking procedure is OK

The condition used here (If all parameters conform to FEL requirements) could itself be subject to a whole series of conditions of its own, such as:

> IF the teeming procedure is correctly designed,
> AND the slag/degassing procedure correctly specified.

This example makes it clear that it could take more than one conditions to prove a rule, and that rule conclusions can be conditions in further rules.

Conditions such as those shown above are always checked in the order in which they appear. In the above example, "The teeming procedure is correctly designed" will always be the first condition looked at. If that condition fails i.e. it is not true, then the second condition is ignored, since both conditions have to succeed for the rule to be proved true and it would be a waste of time checking the second one.

The following are a few typical examples of rules which are dependent on conditions.

Rules governing teeming procedure:-

Rule conclusion: the teeming procedure is correctly designed

Conditions: IF the ingot is over a specific diameter

the aim temp. of steel is $X_{2\text{-}1}$ °C, over the steel T_L

OR the aim temp. of steel is $X_{2\text{-}2}$ °C, over the steel T_L

OR the aim temp. of steel is $X_{2\text{-}3}$ °C, over the steel T_L

Rule conclusion: The aim temp. of steel is $X_{2\text{-}1}$ °C, over the steel T_L

Conditions: IF [Al] is not specified for deoxidising

AND the steel is CrMoV type

Rule conclusion: The aim temp. of steel is $X_{2\text{-}2}$ °C, over the steel T_L

Conditions: IF [Al] is specified for deoxidising

AND the steel is not CrMoV type

Rule conclusion: The aim temp. of steel is $X_{2\text{-}3}$ °C, over the steel T_L

Conditions: IF [Al] is specified for deoxidising

AND the steel is CrMoV type

Rules governing slag and degassing procedures:-

Rule conclusion: The slag/degassing procedure correctly specified

Conditions: Processes for ingot between $X1_a$-$X1_b$

OR Processes for ingot between $X2_a$-$X2_b$

OR Processes for ingot between $X3_a$-$X3_b$

Rule conclusion: Processes for ingot between $X1_a$-$X1_b$

Conditions IF the ingot is between $X1_a$-$X1_b$

AND the steel is required to be VCD treated

AND single slag furnace

AND single slag VAD

Rule conclusion: Processes for ingot between $X2_a$-$X2_b$

Conditions IF the ingot is between $X2_a$-$X2_b$

AND the steel is required to be VCD treated

AND double slag furnace

AND single slag VAD

Rule conclusion: Processes for ingot between $X3_a$-$X3_b$

Conditions IF the ingot is between $X3_a$-$X3_b$

AND the steel is required to be VCD treated

AND double slag furnace

AND 2 single slag VAD/LF degas

Using the above format of rule conclusion and conditions, rules governing the FSL steelmaking procedures have been accumulated and stored in a small expert system. Further rules treated in this way covered the checking of parameters, such as ingot size, ingot weight, type of steel, and so on.

Key words:-

Notice the words AND and IF appearing in the above examples. They are some of the key words in CRYSTAL. In all, there are four key words: IF, AND, OR, NOT, which are used to set up the logical steps in the knowledge base. It is not necessary to include the key words within rule conditions, the actual text of a rule condition is completely free form. The key words are only used to signify the way in which conditions are related.

Commands:-

Commands are used within a knowledge base to carry out special tasks. For example, they can be used to:

Display information for the user;
Accept input from the user;
Print information for the user;
Assign and test variables;
Restart the system;
Give explanations to the user.

Commands can be used in exactly the same way as conditions. Therefore, we can use commands as parts of conditions and ask the user, for example, to input parameters from FSL steelmaking procedure for evaluation by the system rules, or to aid the display of results.

5.2.2 Operation of the Coupling

When it is necessary to activate the approval process for FSL steelmaking procedures, the query sub-system takes control of the overall operation. It swaps the database into high memory or into hard disk, and loads the expert system. The control routine then operates the CRYSTAL system, to interrogate the user, through a series of data entry screen, about the steelmaking procedure that was actually used. This procedure is then evaluated against the relevant set of CRYSTAL rules and a final screen presented to the user indicating whether the steelmaking procedure could be approved or where any discrepancy exists.

5.3 Database Enhanced with External Program

Process engineering applications frequently demand mathematical capabilities that exceed those provided in database systems. Thus a CAPP system based on a database will need to call on programmes written in languages with a considerably greater range of mathematical functionality. In order to explore the nature of the inter-programme links required, a programme was written in C^{++} [S91][L91] is carry out calculations related to hydrogen degassing anneals.

5.3.1 Automation of Hydrogen Degassing Process Design

Hydrogen, above a certain level, causes flaking or internal fracturing in steels. The critical level depends upon the alloy type and the heat treatment condition.

Since flaking depends upon hydrogen diffusion in the cold state it has been found not to occur in items heat treated immediately after forging, with heat treatment cycles designed to allow hydrogen to diffuse out of the product.

At FEL process planning for such hydrogen degassing is normally applied for big forging components or components used in critical applications such as nuclear reactors, power generators or rolls for large rolling mills. The diffusion process in these large forgings is slow and is dependent on the temperature. Provided the hydrogen partial pressure in the heat treatment furnace is sufficiently low, hydrogen will diffuse out of the forgings during any heat treatment cycle: forgings undergo a number of heat treatment cycles during manufacture either to bring them to required forging temperatures or to obtain appropriate microstructural states. Hydrogen diffuses from the forging during all these heat treatment cycles, but their cumulative effect is seldom sufficient to lower the hydrogen content to its final acceptable level. An additional anneal is required specifically to remove hydrogen, but one that does not alter the forgings microstructure. This anneal is termed 'the hydrogen anneal'. Determining the hydrogen annealing time is an important component of the metallurgists' process design activity.

The knowledge that the metallurgists within FEL use to carry out this design activity is expressed in tabular form. The tables are based on simple scientific principles and upon operating experience within the FEL group. The database enhancement system developed and tested in this work did not draw on this operating experience, but carried out a sophisticated scientific calculation, automatically, thus circumnavigating the human activity involved in the use of the tables.

5.3.2 Theoretical Base for Hydrogen Degassing Calculations

We can consider that the forgings are cylindrical in shape and that the rate-limiting step is the diffusion of hydrogen out from the solid in the radial direction. End effects will be extremely small and can be ignored for most practical situations. Hence, we seek a solution to Fick's second law [S63] within an axi-symmetric solid with the held at a constant value.

For these conditions, the Fick"s second law equation is:-

$$\frac{\partial C}{\partial \tau} = \frac{1}{r}\left(D\frac{\partial C}{\partial r} + rD\frac{\partial^2 C}{\partial r^2}\right) \tag{5-1}$$

Where C is the hydrogen concentration in the steel, and D is its diffusion coefficient.

D can be considered as constant, so that:-

$$\frac{\partial C}{\partial \tau} = D\left(\frac{1}{r}\frac{\partial C}{\partial r} + \frac{\partial^2 C}{\partial r^2}\right) \tag{5-2}$$

The boundary conditions are,

$\tau = 0;\ 0 < r < R:\quad C(r, 0) = C_i$

$\tau = 0;\quad r = R:\quad C(R, t) = C_s$

Introducing dimensions $r^* = r/R$, we get,

$$\frac{\partial C}{\partial r} = \frac{1}{R}\frac{\partial C}{\partial r^*} \tag{5-3}$$

therefore,

$$\frac{\partial C}{\partial \tau} = \frac{D}{R^2}\left(\frac{1}{r^*} \circ \frac{\partial C}{\partial r^*} + \frac{\partial^2 C}{\partial r^{*2}}\right) \tag{5-4}$$

The boundary conditions become,

$\tau = 0;\ 0 < r^* < 1 : C = C_i$

$\tau > 0;\quad r^* = 1 : C = C_{eq}$

so we can define a dimensionless hydrogen concentration fractional change,

$$Y = \frac{C - C_{eq}}{C_i - C_{eq}} \tag{5-5}$$

where we get,

$$\frac{\partial C}{\partial} = \frac{\partial Y}{\partial} \circ (C_i - C_{eq}) \tag{5-6}$$

substituting into Eq. (5-3), gives,

$$\frac{\partial Y}{\partial \tau} = \frac{D}{R^2}\left(\frac{1}{r^*}\frac{\partial Y}{\partial r^*} + \frac{\partial^2 Y}{\partial r^{*2}}\right) \tag{5-7}$$

let $X = D\tau/R^2$,

$$\frac{\partial Y}{\partial X} = \frac{\partial Y}{\partial \tau} \circ \frac{\partial \tau}{\partial X} = \frac{R^2}{D} \bullet \frac{\partial Y}{\partial \tau} \tag{5-8}$$

i.e.

$$\frac{\partial Y}{\partial \tau} = \frac{D}{R^2} \circ \frac{\partial Y}{\partial X} \tag{5-9}$$

substitute into Eq. (5-6), we get,

$$\frac{\partial Y}{\partial X} = \frac{1}{r^*} \circ \frac{\partial Y}{\partial r^*} + \frac{\partial^2 Y}{\partial r^{*2}} \tag{5-10}$$

The boundary conditions now become:-

$X = 0; 0 < r^* < 1 : Y = 1$

$X > 0; \quad r^* = 1 ; Y = 0$

$Y = f\{ r^*, X \}$

If we have a mass transfer resistance at the surface, the surface boundary condition becomes:-

$\tau > 0; r = R$:-

$$\frac{\partial C}{\partial r} = \frac{\alpha}{D}(C - C_{eq}) \tag{5-11}$$

Where C_{eq} is the hydrogen mass fraction in equilibrium with the furnace gasses, and α is the mass transfer coefficient.

Making the surface boundary condition dimensionless:-

$X > 0; r^* = 1$;

$$\frac{\partial Y}{\partial r^*} \bullet \frac{1}{R} = \frac{\alpha}{D} Y \tag{5-12}$$

i.e.

When $X > 0$; $r^* = 1$;

$$\frac{\partial Y}{\partial r^*} = \frac{\alpha R}{D} Y \qquad (5\text{-}13)$$

$\alpha R/D$ is a dimensionless number, and the solution is now of the form:-

$$Y = f\{N, X, M\} \qquad (5\text{-}14)$$

Where $M = D/\alpha R$; and $N \equiv r^*$;

This functional relation can be determined by solving equation (5-10) subject to the boundary conditions, Fourier series normally being used to obtain the solution. The mathematics is widely applicable to heat transfer problems, so the Fourier series solutions have been obtained and represented in graphical form by a number of workers. An often quoted set of graphs are the Gurney Lurie charts such as those shown in Figure 5-2 which, on a log/log plot, have the form of straight lines for all but the earliest stages of the diffusion process. It has been shown [S52] that these straight line portions can be represented by equations of the form:-

$$\ln Y = -N_e \circ X + f(N) \qquad (5\text{-}15)$$

where N_e is known as the Newton parameter and is a function of the relative surface resistance M, and of the shape of the solid. It has also been shown that the curves can be expressed in the form of an average fractional hydrogen composition Y_{av}, defined by replacing C in equation (5-5) by C_{av} , and that the variation of this average fractional

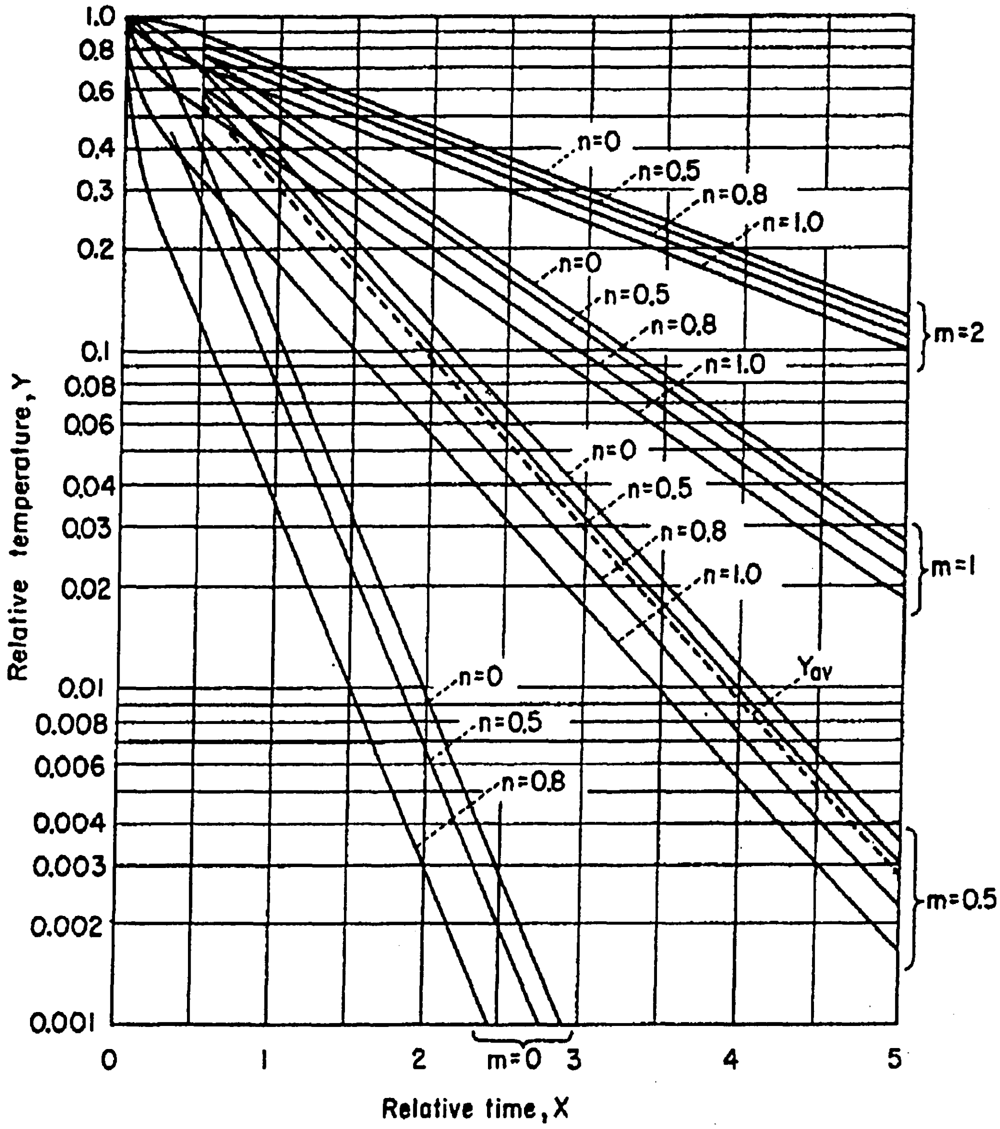


FIG. 5-2 Gurney-Lurie Chart for Infinite Components

composition is represented by the equation:-

$$\ln Y_{av} = -N_e \bullet X \qquad (5\text{-}16)$$

Schuhmann [S52] includes a graph of N_e values against M which is reproduced here in modified form in Figure 5-3 and shows that N_e tends towards constant values as the surface resistance becomes vanishingly small. The limiting value of N_e for cylinders is 5.75, so that equation (5-16) becomes:-

$$\ln Y_{av} = -5.75X \qquad (5\text{-}17)$$

Since hydrogen diffusing out of the forgings during heat treatment will react with the furnace gases to be rapidly transported away from the surface, the surface resistance to hydrogen removal is zero, and equation (5-17) can be used to calculate diffusion time.

Given a target value of the average fractional concentration:-

$$[Y_{av}]_{target} = \frac{[C_{av}]_{target} - C_{eq}}{C_i - C_{eq}} \qquad (5\text{-}18)$$

we can substitute into equation (5-17) to obtain a target value of X_{target} and hence a target value of diffusion time, τ_{target}, where,

$$\tau_{target} = \frac{R^2 \bullet X_{target}}{D} \qquad (5\text{-}19)$$

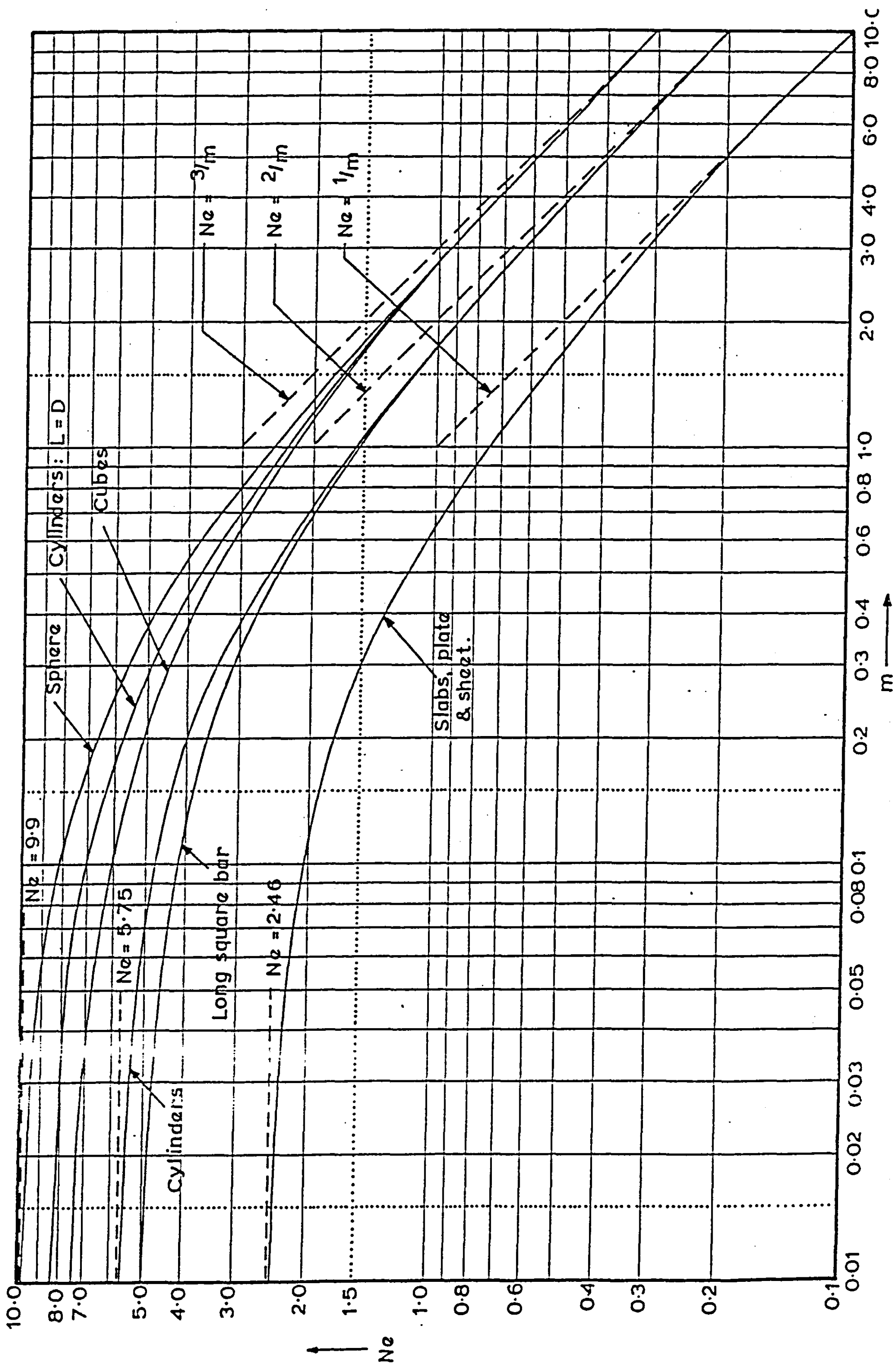


FIG. 5-3 Ne vs. Relative Surface Resistance (m) for Single Geometric Shapes.

substitution equation (5-19) into (5-17) gives:-

$$[\tau_{av}]_{973K} = -\frac{R^2}{D}\frac{Ln[Y_{av}]_{t\arg et}}{5.75} \tag{5-20}$$

However, in the hydrogen annealing process, we are concerned with controlling the maximum hydrogen content which is in the centre of the component. Schuhmann has further shown that the straight line portions of the Gurney-Lurie charts can be represented by equations of the form:-

$$\text{Ln Y} = -\text{ Ne }(X - \lambda) \tag{5-21}$$

where λ is a log factor which depends upon position [S52]. Since the centre line composition of the forging always lags behind the average value, Y is a positive value, and the value Y at centre position can be found by using the values of Y/Y_{av} interpolated from Figure 5-4,

$$Y_c / Y_{av} = 2.3 \tag{5-22}$$

Therefore the centre and average hydrogen content have the following relation,

$$\frac{C_{centre} - C_{eq}}{C_i - C_{eq}} = 2.3\frac{C_{av} - C_{eq}}{C_i - C_{eq}} \tag{5-23}$$

If we assume that oxidising conditions are maintained in the heat treatment furnaces, we

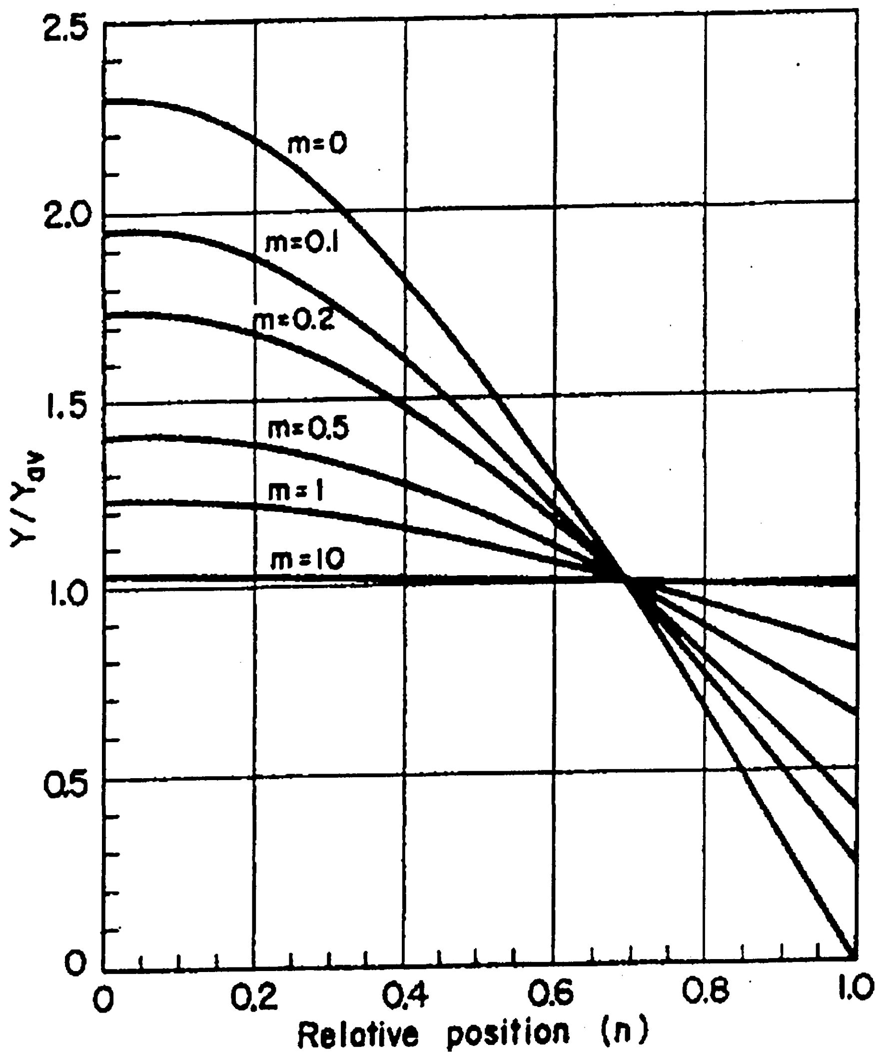


FIG. 5-4 Long Cylinders: Variation of Temperature with Position for Different Values of Relative Surface Resistance.

have that $C_{eq} = 0$, so that,

$$\frac{C_{centre}}{C_i} = 2.3 \bullet \frac{C_{av}}{C_i} \tag{5-24}$$

Therefore,

$$\left[\tau_{av}\right]_{973K} = -\frac{R^2}{D_{973K}} \frac{Ln\left[\dfrac{C_{centre}}{2.3C_i}\right]_{target}}{5.75} \tag{5-25}$$

Where the temperature 973K (700°C) has been taken as a reference temperature. The target time at a different temperature, θ, would be different, but would follow the following relation:-

$$D_{973K} \bullet \Delta\tau_{973K} = D_\theta \bullet \Delta\tau_\theta \tag{5-26}$$

Given a target centre line hydrogen concentration $[C_{centre}]_{target}$ and a known initial level C_i, measured when the ingot was cast, we can calculate the required diffusion time at the reference temperature using equation (5-25).

The contribution towards this target time from an anneal at a different temperature θ can be determined, as shown by equation (5-25) from the equality (5-26).

If a component undergoing a complex heat treatment schedule involving a number of different temperatures such as the one shown in Figure 5-5, the length of the hydrogen

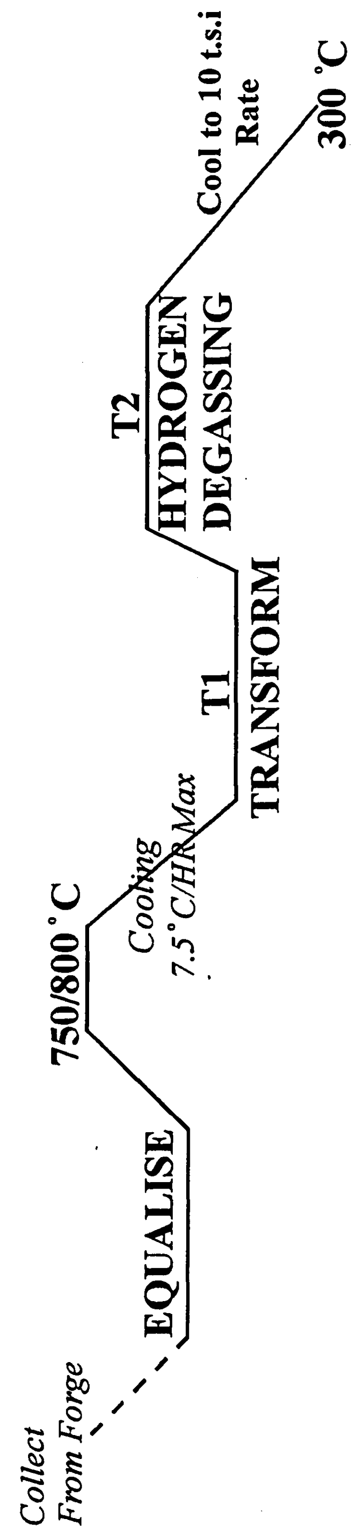


FIG. 5-5 Heat Treatment Cycles for Component going through Different Temperature Stages

anneal can be calculated by adding together a number of equations of the form of equation (5-26):-

$$(D_{973K} \bullet \Delta\tau_{973K})_{req} = D_{\theta_1} \bullet \Delta\tau_{\theta_1} + D_{\theta_2} \bullet \Delta\tau_{\theta_2} + D_{\theta_3} \bullet \Delta\tau_{\theta_3} + \ldots + D_{Degas} \bullet \Delta\tau_{Degas} \quad (5\text{-}27)$$

gives the time required for hydrogen degassing as:-

$$\Delta\tau_{Degas} = \frac{(D_{973K} \bullet \Delta\tau_{973K})_{req} - D_{\theta_1} \bullet \Delta\tau_{\theta_1} - D_{\theta_2} \bullet \Delta\tau_{\theta_2} - D_{\theta_3} \bullet \Delta\tau_{\theta_3} - \ldots}{D_{Degas}} \quad (5\text{-}28)$$

The chemical diffusion coefficient varies with temperature, the following equation being given in Smithells Metals Reference Book [BB92] as:-

$$D = 0.93 \times^{-3} (cm^2 s^{-1}) \exp\left(-\frac{2.7(kcal \bullet mol^{-1})}{R\theta}\right) \quad (5\text{-}29)$$

The time required for the DEGAS heat treatment cycle at any specified temperature can be calculated using equations (5-28) and (5-29) if we know the temperatures and time of the other specified heat treatments.

A program listed in the appendix 6, has been designed using the C^{++} programming language to calculate hydrogen degassing time according to this model.

5.3.3 Architecture of the Experimental System

As shown in the Figure 5-6, there are three major components in the current experiment:-

* Database Relations
* Database Query Sub-System
* External Program

Database Relations

The heat treatment process data are stored in the database relations. There are three relations- HTP (Heat Treatment Procedure), HTP_Status and HTP_Comments. The relation HTP is used to identify the heat treatment with key fields of the procedure number (HTP_No.) and issue letter (HTP_I). Each tupple of the heat treatment records the designer and the date with which the plan is issued. The relation HTP_Status contains a group of forgings (FEL order numbers) which are going through the same heat treatment procedure. The relation HTP_Comments contains the information on the sequence of the heat treatment. At each stage of the heat treatment, the information on process name (Stage), heating or cooling rate (Heating rate or Cooling rate), temperature (Temp) and holding time (Hold) are given in this relation.

Database Query Sub-System

If hydrogen degas process planning is initiated, the database query sub-system will collect information about the component features and their design parameters from the relevant process planning areas. For example, the information about heat treatment cycles are acquired from the Quality Plan planning area, the information about the ingot

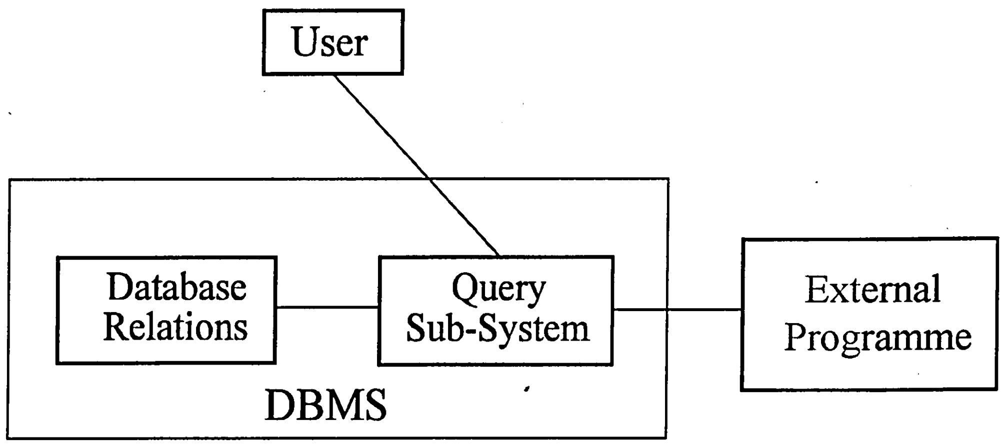


FIG. 5-6 Enhanced Database System: External Enhancement

and cast analysis, such as ingot diameters and hydrogen level, from the Steel Order planning area. The information is downloaded into a DOS file, HYDRO.DAT, in fixed length ASC II format, which can then be read by a DOS resident programme. Based on this information gathered the external programme can generate the length of hydrogen degassing period at required temperature. The result will then be imported and put into the heat treatment procedure accordingly by the query sub-system.

External Programme

The external programme is written in C^{++} has been compiled into a DOS executable file as shown in appendix 6. Therefore, the external programme can be called by the query sub-system within the database system. The programme contains several functions which are used to achieve the final solution. The first function in the programme reads all the component features and their design from HYDRO.DAT. The second function calculates the degassing effect of each stage of the heat treatment used to achieve certain internal microstructures. The last function calculates the degassing time required for achieving a target hydrogen level. The calculated result is then displayed to the user and saved in a DOS file, F-RESULT.DAT, in ASC II format, which will be imported by a procedure within the database query sub-system.

Concurrency Control

The query sub-system plays an important role in the hydrogen degassing process planning, by not only transferring information between the relations and the programme, but also providing a mechanism of concurrency control.

The concurrency control is for co-ordinating access by simultaneous users so that each user is insured access to a consistent state of the database. If the external programme is to be called the user's name and the relevant process plan identity will be checked by the query sub-system against records stored temporarily in a relation. If this checking is satisfactory, the information will be recorded in that relation and the external programme can be run. The system thus only allows users to access the external programme for a specified heat treatment procedure in order to prevent a conflict. The record will be deleted after the query sub-system has imported the relevant data and put the final result into the specified heat treatment procedure.

5.3.4 Operation of the System Enhanced with Programme

When the query procedure is called by a user, a block data entry form will be displayed into which the relevant heat treatment process plan identity can be entered. The procedure will, first of all, check the entry with the heat treatment records to see if the right one has been specified and then check if any other user is using the external programme. If these checks are satisfactory, the procedure will collect and export the heat treatment cycle records such as process name, aiming temperature, heating or cooling rate and duration time to be taken at that aim temperature into the DOS ASC II file, HYDRO.DAT. The data can be read by the programme and are necessary for the programme to accumulate the effect of hydrogen degassing at each heat treatment cycle and present a final result to the user. The final result saved on the DOS ASC II file, F-RESULT.DAT, can then be imported from an external file and put into that heat treatment process plan by the query procedure. A typical example of this process for a heat treatment is shown in Figure 5-5 where the heat treatment cycles for a component going through different temperature stages are illustrated.

5.4 Summary

The two enhancement methods to the database system have been explored and discussed in this chapter. The database coupling method with an external expert system has been applied to the approval process for FSL steelmaking procedures which had previously been solved using the database system. Enhancing the database with an external programme has been used to solve complex mathematical calculations for the database system. Both approaches will be discussed further in chapter 7, where issues related to both approaches will be discussed.

CHAPTER 6

OPERATING EXPERIENCE

6.1 The FEL CAPP System

The majority of this research work has been involved with the development of the database system. The system has up to 100 forms and 120 procedures. It was developed under DOS on a Novell computer network, with three diskless 386 computers running as intelligent terminals, one for development work, one for the metallurgists to use in tests and one for the QA personnel. The Novell network served all the computing needs at FEL such as accounting, order enquires, word processing and some drawing packages, leading, at times, to a long response time. The three diskless computers used in this work had the conventional base memory of 640K and extended memories of 384K. The database software used was DataEase, version 4.0, which is not able to use extended memory but is able to use expanded memory, although none was available on the three computers.

The FEL CAPP system was operated by the metallurgists in tests to produce manufacturing plans, steel orders and quality plans. Facsimile examples of the output from this activity, which do not represent real plans for real forgings, are shown in Figures 6-1 to 6-2. This output was incorporated into the set of documents for each forging identity that forms the kernel of the established paper based planning system that is operated by FEL under a practice built up over a number of years.

The metallurgist did not consider the computer system to be an advantage over the paper based system. In the first place, they were required to enter data related to specifications into the system through the keyboard whereas the manual system merely required them to copy the data by hand into a proforma. Proformas existed in the paper system, moreover, for manufacturing plans, also merely requiring the metallurgists to tick boxes for the operations that were to be included. A similar facility was provided in the

FORGEMASTERS ENGINEERING LIMITED
MANUFACTURING PLAN *ISSUE: A*

FEL Order No.(s)	5447944/5 Sales No. E/21212 Date: 02/10/92
Customer:	Power Generation Co.
Inspection:	None
Description:	Ring: 143.00" O. D. 1300.00" I. D. 36.00" Face
Material Spec.	AISI 4340 H (Mod.)
Other Spec(s):	Cust. Material Specification for Forged Rings.
Drawing No.	Cust. Material Specification for Forged Rings.

Analysis Range Required: RMC Code: 6817/06 (Modified) Scrap Code -- Solid: NC Turning: NT

	C	Si	Mn	P	S	Cr	Mo	Ni	V	Al	Sb	Sn
Min.	0.25	0.15	0.55		0.005	0.40	0.40	0.40	0.10			
Max.	0.36	0.20	0.85	0.010	0.015	0.45	0.45	0.60	0.14	0.006	0.004	0.010
Aim				ALAP	ALAP							

	Cu	N	As	Ti	Nb	___	___	___	___	H2 (ppm)	Note
Min.											
Max.			0.020							1.4	Wt% Unless Indicated
Aim										1.0	

Special Steelmaking Requirements	*Carbon eqivalent (C. E.) < 2.0* *C. E. = C +Mn / 6 + (Cr + Mo + V)/5 + (Ni + Cu) / 10*

OUTLINE MANUFACTURING PROCEDURES

OP No	Operations	QA. Doc.	Comments	QA	Prod Time
1	Cast Ingot				*137*
2	Forge	WOP E410			*138*
3	Anneal / Degas	WOP E522	to be followed immediately by N&T.		*141*
4	Normalise&Temper	WOP E515			*141*
5	Descale&Mark out	WOP E515			*141*
6	Rough M/C	WOP E522	250 CLA max. Allow 1" per surface on all		*143*
			surfaces. Round off corners with 3/8"		
			min. redial.		
7	Dimensional	WOP E714	FEL requirement.		*144*
8	U / S Test Cat. |2|	WOP E714	FEL requirement. NDE 591 Rev. 0.		*144*
9	Water H & T	WOP E714	950 deg. F min. tempering temperature.		*147*
10	Hardness Test	WOP E704	FEL requirement. 4 spots 90 deg. apart at		*148*
			each end.		
11	Mech. Test	WOP E704			*151*
12	M / C for Despatch	WOP E704			*152*
13	Dimensional	WOP E704			*153*
14	Visual Exam.	WOP E704			*153*
15	U / S Test Cat. |3|	WOP E704	ASTM A388, test in longitudinal direction.		*153*
16	Document Review	WOP E704	Certification of chemistry, thermal		*153*
			processing, U/S test, mechanical properties.		
15	Despatch	WOP E704			*153*

Test Unit & Test Req'd Per Unit	1. RT Tensile 2. Charpy R. T. 3. Jominy 4. Micro	*Each Forging* *Each Forging* *Each End* *Each Forging*
Test Position	*Tangential. Sub-surface.*	Quality Assurance Specification: *1. None Quoted.*
COMMENTS	*Tensile and Charpy results for information only.* *Hardness: 303/401 BHN (on forging).* *Jominy hardenability: Rc 47 min. at J 32/16.* *Microcleanness: See section B of H&S Material Specification for* *Medium Carbon Alloy Steel Rolled or Forged Rings.*	

FIG. 6-1 System Output of Manufacturing Plan

STEEL ORDER NO.	000155 -- A
TO PRODUCER:	FSL
INITIAL:	A. J.
DATE:	19/12/92

FORGEMASTERS ENGINEERING LTD
P.O. BOX 287, BRIGHTSIDE LANE
SHEFFIELD S9 2RW
TELEPHONE: 0742 449071
FACSIMILE: 0742 422103

FEL Order No.	Customer:	Description: Generator Rotor. Application: "A" Spare Rotor	Quality:	RMC Code:	Cost/Tonne
5607315/26	Power Generator Co.		3 1/2% CMV	9917/02 (Modified)	

FEL Melting Range Required: (Wt % Unless indicated)

	C	Si	Mn	P	S	Cr	Mo	Ni	V	Al	Sb	Sn	Cu	N	As	Ti	Nb	—	—	—	—	H2 (ppm)
Min.	0.25	0.10	0.15		0.005	0.40	0.40	3.40	0.10													
Max.	0.30	0.20	0.25	0.010	0.015	0.45	0.45	3.60	0.14	0.006	0.004	0.010	0.10	0.020	0.020							1.4
Aim				ALAP	ALAP	0.60																1.0

Acceptable/Customer Requirements: (Wt % Unless indicated)

	C	Si	Mn	P	S	Cr	Mo	Ni	V	Al	Sb	Sn	Cu	N	As	Ti	Nb	—	—	—	—	H2 (ppm)
Min.	0.22	0.08			0.005	0.30	0.35	3.30	0.08													
Max.	0.32	0.20	0.40	0.015	0.018	0.70	0.50	3.70	0.15	0.010	0.006	0.014	0.14		0.025							

Note / FEL Requirements: (Hot To Forge: *Yes*)

No. 6. A good stream degas is essential. Low Residual scrap to be used. Normal strip.

Customer Requirements:

1. Vacuum Degassed: *Yes*
2. Deoxidation: *Not Specified*
3. Teeming: *Not Specified*
4. Ultrasonic: *<=2mm FBHE*

Documentation Required

1. Chemistry (Fax): *Yes*
2. Melt Record: *No*
3. Cert. of Conformity Inc. Analysis: *Yes*

INGOT REQUIREMENTS SIGNATURE

GROUP	FEL ORDER NO.	SIZE		QUANTITY	WEIGHT (T)	DELIVERY	TCD	INITIAL	DATE
A	5607315 5607317	X1 M/F length(in.)	100.50	2	53.14	10000	W12	K. A.	19/01/92
B	5607316 5607318/9	X2 M/F length(in.)	100.20	3	50.00	10000	W23	K. A	25/01/92
C	5607320/22	X3 M/F length(in.)	51.00	1	21.42	10000	W12	K. A	25/01/92
D	5607323/26	X4 OCT length(in.)	81.50	1	10.78	10000	W15	K. A	25/01/92

FIG. 6-2 System Output of Steel Order

computer FEL CAPP system, which also supported the re-use of previous successful plans for repeat identical orders. The advantage that these facilities could provide was negated, however, by the slow response of the network server's disc system when recalling past plans.

The development work on the FEL CAPP system was carried out through consultation with the metallurgists and this exposed them to operating malfunctions which inevitably existed in the earlier experimental and development versions. This exposure undermined the metallurgists confidence in the reliability of the computer based system in comparison with the paper based system.

The quality assurance functions concerned with document distribution and return was operated by the quality control department with considerable success, and is now an established component of FEL's procedures.

6.2 The Enhanced Database System

The tests of the enhancements to the FEL CAPP System with an expert system and with an external programme were conducted at Sheffield Hallam University. A database environment similar to that developed at FEL was created in order to conduct tests on the operation of the enhanced FEL CAPP System. Both enhancements were developed and tested on a 286 computer with the conventional base memory of 640 K and extended memories of 384 K. Tests have also been conducted on a 486 DX50 computer.

The enhancement of the database system with an expert system was limited to the approval of the steelmaker's proposed teeming and degassing procedures. It was not extended to checking the steelmaker's planed melting range against the range specified by FEL in the steel order. This system was successful in assisting the user to analyse the FSL steelmaking procedures, but it was difficult for the user to add to or modify rules, as approved practice developed, because the rules were structured in the CRYSTAL expert system. The system that was developed, and contained 100 rules and 184 commands, could be run in a reasonable fast speed. Because the expert system was run from within the FEL database system, the major limit to the speed of operation was the time taken for DataEase to swap itself to either hard disk or extended memory and to load the expert system. Obviously, on a 486 machine the test would be faster than 286 machines. The output of the system is shown in Figure 6-3.

The enhancement with an external programme designed in C^{++} has achieved distinct results which are difficult to achieve by the methods discussed so far. Because C^{++} is an objected-oriented programming language, allowing the design of clearer, more reliable, and more easily maintained programs, it is easy to structure the programme by grouping and linking different functions that carry different specific tasks. Once the programme is called from the FEL database system, the final result can be reached within a second, because DataEase can run external programmes directly as if they were internal query procedures. The output of the system is the screen display of the final result as shown in Figure 6-4, and this result can be transferred back to a relevant heat treatment procedure in the FEL database system.

CRYSTAL EXPERT SYSTEM

STEELMAKING PROCEDURE APPROVAL EXPERT

The Steelmaking Procedure has been Approved by the Expert System as Shown below:-

1. The Over Heat Temp. is OK.
2. The Furnace Process is OK.
3. The Casting Position is OK.
4. The VAD Process has been specified.
5. The Spare Metal is Sufficient.

The action to be taken: {Restart} or {Print}

Select a choice by moving the cursor left or right

Press Enter When Ready

FIG. 6-3 Expert System Output of Approved Steelmaking Procedure

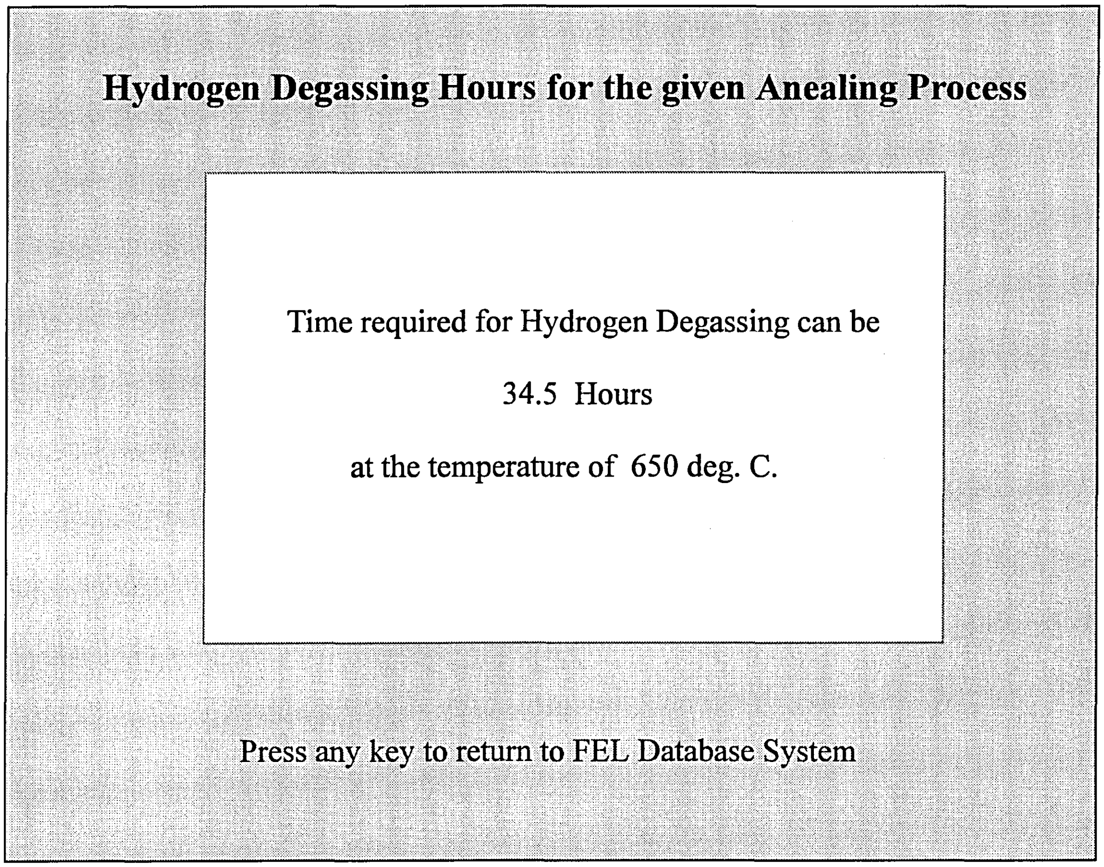


FIG. 6-4 System Output of Calculated Hydrogen Degassing Hours

CHAPTER 7

DISCUSSION

7.1 Introduction

The existing CAPP system has demonstrated that Database technology can be successfully applied to manufacturing engineering problems in general, and applied to the domain of metallurgical process planning for the manufacture of large steel forgings in particular. Two hierarchy control levels have been adopted to organise development of the process plans and to maintain the supporting engineering data. This structure is essential to the management of large amounts of versional manufacturing data involved and for the effective incorporation of manufacturing knowledge into the system.

7.2 Versional Approach

Since a number of different plans are being designed and then modified during their execution, a versional database approach, as advocated by Blanken and Ijbema [BI87], has been adopted. A further development of this versional approach has involved the creation of distinct planning area within which separate component parts of the overall planning process are conducted. The versional approach is not applied to the development of the individual sub-plans in each separate planning area because this development is normally conducted by a single planner rationalising a restricted sub-set of design decisions. The versional approach must be applied, however, if a sub-plan has been issued and decisions involved in its creation must be incorporated into planning in the other planning areas. This is true within the two hierarchy control levels in the current CAPP system, and also at the third level, concerned with the operation of process and process plans, which has been excluded from the current scheme.

A strategy has been adopted involving the use of facsimile files that have a local existence only in the individual planning areas. These facsimile files allow planning to proceed locally, without affecting the structure of versional files that have been issued to support general decision making purposes. Planning decisions can be made locally without affecting or invalidating planning decisions made in other planning areas, some of which exist outside the CAPP system developed in this work.

Once planning is completed in the planning area, the records comprising the sub-plan are **'issued'** - that is they are placed in the main data files of the database and the facsimile records are deleted. The issue of a sub-plan makes it available for consultation and data abstraction in the other planning areas within the database and for the planning of production schedules at plant level etc. Once a sub-plan has been issued and planning decisions in other areas based upon it have been made, the sub-plan can not be withdrawn in the event of its requiring modification. Thus it must remain as a clear record identity in the database system and the modification made to it must be stored as a separate but related identity. This has been achieved through the use of a double field identity key structure in all parts of the database. Each set of versional sub-plans is identified by a single value in one field, and by a variable value in the second field changing from version to version. The first field is normally a numerical field, the second field normally a single alphabetical field.

Although this versional approach was adopted from the outset in order to maintain data integrity within the database, it was found to model almost exactly the manual system that had been set up within FEL to meet quality control requirements. It was a model, however, that used a great deal of computer file space since all the data for each version was stored in a separate record. Since the creation of a large number of records in this way could extend time spent in data search, an alternate strategy was contemplated in which only altered data need be stored for each revision. Such a strategy would require each versional records to store the identity of the field that had been altered as well as the altered value, but not unaltered values. It was not, however, possible to appraise whether this method would save time on data search because DataEase, the database management system used in this work, would not allow a variable field name to be used in a procedure instruction.

The issue of a revised (new version) sub-plan within the CAPP system is normally required because of decisions made outside the sub-plan's home planning area. These decisions are either internal to the forging manufacturing process, or external to it. External decisions typically arise when a customer revises some aspect of the original design requirement, this type of revision being normal practice in the manufacture of one-off complex engineering plant through a system of main- and sub- contracting firms. Decisions made internal to the forging manufacturing activity arise when processing at a certain stage has not met the requirements placed upon it. Such an occurrence is subject to a discrepancy report and regular management sessions review these reports to decide which will require remedial action, such decisions often involving consultation with customers. Remedial action, if required, will normally involve revision of planning in one of the sub-planning areas, leading to the issue of new versions of sub-plans.

One aspect of the versional approach has had a major impact on the design of the database, and this stems from the way in which ingots are grouped within FEL for process planning purposes. Forgings are frequently ordered in batches to meet the same specification and therefore all require the same manufacturing plan. Thus forging definition records, manufacturing plan records etc. normally apply to a group of forgings. This group is normally specified by filling a key, or foreign key, field in the database with a sequence of numbers, see section 4.3.1.1. Changes in design requirements are often applied to single forgings such as a group, or to a restricted sub-set of the group, and this is also the case of discrepancy reports for which remedial action must be taken. A system of 'pointer' or status relations was therefore included in the database, as shown in Figure 4-5. Individual records in these files keep track of the relationship between the different versional designs and the subset of the FEL orders to which they apply. This was necessary since planning decisions made at different stages of the manufacturing process were applied to different sub-groups of the forgings within an original forging definition record.

7.3 Authorisation Control

By using query sub-system, the different process plans are integrated into a single system rather than isolated as separated entities within the planning system. When a process plan is being designed, in one planning area, related manufacturing data from other areas can be obtained and presented through the relational links within the overall planning system. Constraints have been built to the routine programmes and procedures that enable the manufacturing knowledge from one area to be presented in another planning area whilst guarding against un-authorised deletion and modification.

This protection was partly managed using the security system within DataEase which allocated a security level to each user. This system controlled the read and write access that the different users had within the database, but suffered from the major disadvantage that it was a global hierarchical system within the database as a whole. It would have been more flexible and easier to develop a secure system if DataEase had allowed different hierarchical security structures to be set up for individual files or groups of files. This would have made it easier to retain security in the planning processes in each of the sub-planning areas. The use of facsimile files for local planning processes, however, was adopted, least in part, to overcome this disadvantage.

7.4 Incorporating Metallurgical Knowledge

A considerable amount of specialist manufacturing knowledge has been incorporated into the database. Although the system uses a database management system, the incorporation of this knowledge allows it to be considered as an intelligent CAPP system. Specific points where metallurgical knowledge has been incorporated include:-

1. the relationship between chemical specifications, mechanical property requirements, and heat treatment schedules. The chemical specifications include those authorised by National standards bodies - BSI, DIN etc. - International bodies - ISO standards and specifications developed by individual customers. Some customers do not have the knowledge to specify the chemical specification, merely specifying the mechanical properties required, or even merely the function for which the forging is to be made.

2. the relationship between the chemical specification, the rationalised melting codes (RMC), the steel making processes, the heat treatment schedule and the mechanical property range required. The decisions involved tightening specification ranges for key chemical elements, and specifying certain steel making procedures. These decisions are not based purely on technical criteria alone since the cost of different steelmaking procedures can vary significantly.

3. the relationship between the RMC codes and the scrap codes. These relations have been built up over many years of manufacturing special steels and are adopted to minimise raw material costs whilst maintaining safe levels of impurities and tramp elements.

4. the approved steelmaking procedures. Currently the steel making procedures adopted by the sister company, FSL (Forgemasters Steels Ltd.) for ingots of a specific diameter, require approval within FEL at Director level before steelmaking commences. This is merely a data checking activity comparing a proposed set of procedures with an agreed set of manufacturing rules. These rules have now been incorporated into the CAPP database system using a set of rule templates into which approved sets of values can be read from data records.

The manufacturing knowledge described above has been stored in the database as a set of historical decisions, or approved rules based on historical experience. These historical decisions can then be presented to the process planner as the basis on which to make current decisions. As such the CAPP system developed is a variant system. There are three basic reasons why a variant systems has been adopted which can be classified as a economic reason, a social reason and a legal reason.

From the economic point of view, two metallurgical process planners are involved on a day to day basis within FEL in the planning of metallurgical process plans. These process planners are applying an extremely complicated set of domain knowledge that would be highly expensive to encapsulate in a generative CAPP system. As far as the manufacture of large specialist steel forgings is concerned, FEL has 5 or 6 competitors on a world-wide basis so that there could never be financial justification for the high cost that would be involved in creating a generative system. Not only would the cost of developing a generative system by prohibitively high, but so also would be its maintenance. Design requirements for steel forgings change steadily so that a generative system would have to undergo continual modification. A further aspect of the financial argument stems from this restricted nature of the large steel forging industry. With so few competitors, the commercially efficient use of manufacturing knowledge is one of the principal factors on which each firm's competitiveness depends.

The sharing of that knowledge is perceived by each firm to be a major attack on its competitiveness.

7.5 Variant CAPP System Approach

The social reason is not unrelated to the economic reason. It is obvious that the metallurgical planning of specialist steel forgings is a highly specialised activity and that planning expertise is therefore highly praised on an individual basis by the planners. Collecting domain knowledge for incorporation into a generative system would therefore be resisted by the planners, and the financial basis would not provide sufficient argument for that resistance to be overcome. The variant system has, however, been given a limited welcome by the users, especially in the way in which steel order planning is assisted by the computer search through the rational melting codes used in related and previously approved orders.

The legal reason why a variant system has been developed concerns the structure of responsibilities taken within the main and sub-contractor structure of the heavy plant manufacturing industry. We have already seen that the approval of FSL steelmaking plans - FSL is a sub-contractor to FEL - for ingots over a specific diameter must be made within FEL at Director level. FEL itself is, of course, producing forgings as a sub-contractor to other heavy plant manufactory companies. Responsibility is being taken for extremely large processing and raw material costs and it is a requirement within the contract structure that this responsibility is taken, at least in FEL's case, at Director level. Although the computer could perhaps assist and advise the director in taking the decisions involved in the discharge of that responsibility, it appears unlikely that the legal structure would ever allow those decisions to be made by a generative CAPP system.

7.6 Knowledge Development Process

The knowledge development process represents the capturing concept of the engineering knowledge. Figure 7-1 shows the way in which the system gains engineers' experience through the design process. As already mentioned in previous chapters, knowledge, in the form of rules and data, is structured in the database through the relationships between forms and procedures. The system assists users in the development of process plans, but it also learns from the users and stores data input by the users in a form that can be recalled later.

An important aspect in this knowledge developing process is that, as shown in Figure 7-1, present engineering knowledge is stored in temporary form during the design process, to which can be added past engineering knowledge through feedback. This is essential to assure long-term stability of information systems. Feedback information is always valid in terms of matured knowledge and company rules. This ability of the system to retain past manufacturing knowledge and to submit it to the current design process enhances long term stability of the manufacturing system. Knowledge in the form of previous decisions clearly tends to stabilise the system. It does not, however, enhance current decision making unless information is also available about the constraints operating when the decisions were made, and the extent to which the decisions were successful.

This process allows engineering knowledge to be updated each time it is used, and is the main reason why facsimile files are adopted for the design process. The facsimile files allow the design process to be separated from knowledge storage. They also allow all necessary constraints to be checked and/or applied before storing the knowledge in the database.

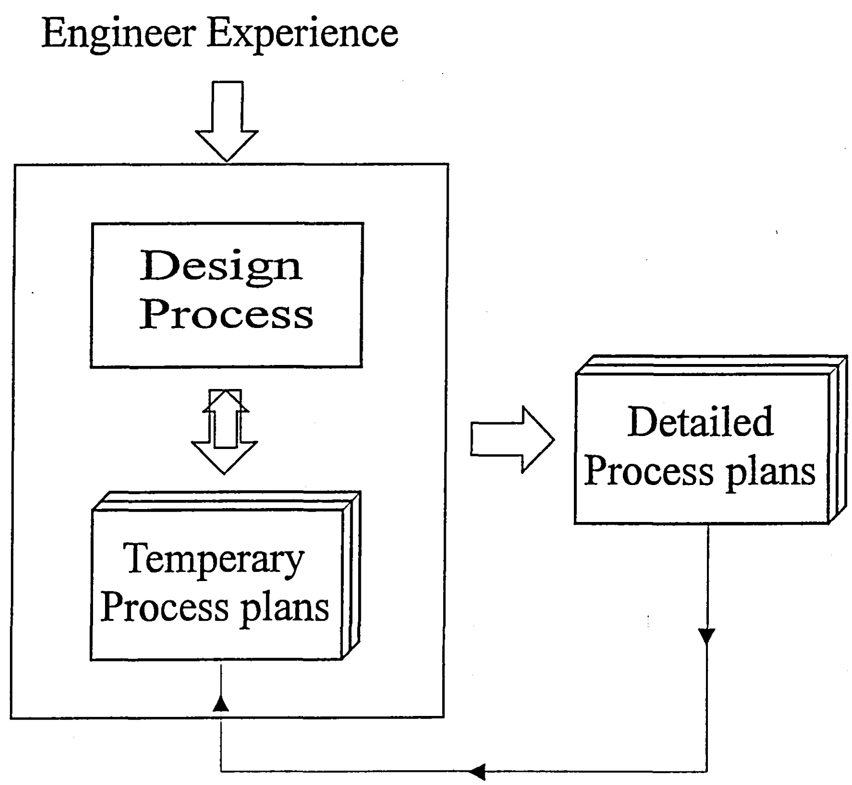


FIG. 7-1 Knowledge Development Process

7.7 Hierarchical Process Planning

Process planning can be viewed as a decision making activity which generates manufacturing instructions for the conversion of a part design into the finished product. The input variables include the design specifications and large volumes of manufacturing knowledge and process capability. The final output includes detailed descriptions of the selected processes, machines, process sequences, and quality control status.

The hierarchy used in the FEL CAPP is described in Chapter 3. The system has been organised so as to decompose the process planning process to achieve system efficiency, the manufacturing knowledge and rules being collected into individual planning areas. In each individual area, the process plan is generated according to rules and manufacturing constrains within that area. This minimises the time spend in searching for rules and matching them, activities on which a global rule based system could spend 80% of its run time [F82].

Each time a new plan is created it remains in temporary form until it is ready to be issued (or available) to other sub-planning areas in the system. Therefore, at the highest level in this hierarchy, design features are selected to convert the initial material into the desired product. The manufacturing plan and quality plan are the main activities that dominate the subsequent process planning. At this stage, the specifications are well maintained for each contract order. Although the number of forging definition records is minimised by grouping together similarly shaped forging from a single contract, clear identification is retained through use of the Forging Status file. This minimises the data handling task, whilst allowing tight control to be exercised over the manufacture of each forging. At the highest level, also, the links are established between the forging specifications and the forging definitions to which they apply, and these links are well protected through the subsequent planning process.

7.8 Comparison of Rule Templates vs. Expert System

Both the rule template model and an expert system have been used to represent the manufacturing knowledge involved in the approval of FSL steelmaking procedures. Knowledge governing the approval of teeming and degassing procedures was incorporated into a Crystal expert system comprising 100 rules and 184 commands. The same knowledge was incorporated into the database template model using a main form, containing 4 tupples, supported by a sub-form containing 12 tupples, the form and sub-form being manipulated by two procedures. The database template model used considerably less memory space than the expert system, and possessed a further advantage in that is was the more easily extended to carry out melting range checking.

As described in Chapter 5, the steelmaking Company, FSL, would propose its own melting range in the light of its knowledge of steelmaking processes and cost. Needless to say, the most stringent requirements placed on this melting range is that it should lie within the melting range specified by FEL in the steel order. Typically, a melting range would comprise the maximum and minimum concentration for 18 set elements and 4 user defined elements. The numerical checking of one melting range against another could be carried out very simply in the database system, even though each range was stored in a separate database, because the relevant forms in the two database had the same structure. To carry out the same checks in the expert system requires complex procedures, for exporting the data from the database into the expert system, and then an extensive set of rules to carry out the magnitude checks. A further advantage for the database system was that the way in which it represented melting ranges could the more easily be made to resemble the paper system with which the metallurgists at FEL had considerable experience.

A further advantage of the database rule-tupple system for representing manufacturing knowledge is that it could more easily and more safely be modified as manufacturing knowledge and approved practices evolved. Any particular manufacturing practice is represented by a combination of field values in a rule-tupple. A change in practice would be represented by a different approved combination of values in a new rule-tupple. The creation of such a new rule-tupple can very easily be achieved and the process protected so that it can only be carried out by users who have the appropriate level of security. Such a system of instigating changes in manufacturing practice can also be considered "safe" because a previous practice can still reside in the system as a rule-tupple that exists but is not in current use - using a type of versional approach. Thus, if changed circumstances dictate that the previous practice should be reinstated, this can be done very easily by activating the previous rule-tupple.

These steps are not so easily carried out in an expert system where the manufacturing knowledge is represented by interactions between a complex set of rules. A change in manufacturing practice requires that the current interactions are replaced by a new set, a process that requires the rewriting of some, or all, of the rules. Such a rewriting process cannot be carried out easily and could result in unforeseen anomalies being left in the rule base; to ensure the security of the result would not be straight forward. Moreover, since the approved knowledge lies in the interactions between the rules in the rule base, there is no room for rules that described the previous practice. Changing practice between current and previous successful practice as circumstances change would be fraught by many difficulties.

There is also a further possible advantage of the database system related to the establishment of manufacturing discipline. Modern manufacturing quality control systems require that procedures are carried out repeatedly within tightly controlled limits. Such tightly controlled procedures can be clearly manifest in rule-tupples that specifies these limits. They can, of course, also be incorporated into a rule base in an expert system, but their visibility is not so high. This could make it difficult to check that manufacturing procedures described in a complex set of rules comply with tight specifications.

Through this attempt, we can see that database system approach is the most convenient when limited inference capability is all that is needed and/or there are large masses of data to be handled. It is interesting to note that Ramirez [RD90] who considered various modes of coupling expert systems to database management systems concluded that obvious advantages lay with the use of a database alone.

7.9 Evaluation of the Use of an External Programme

For many engineering query system within applications, it is unlikely that the database engine can handle the calculation problems involved, so that these need to be referred to an external programme. Such calculations require a greater range of mathematical functions than is available in the engine, particularly for manipulating arrays. Typical arrays are those require to store time, heating or cooling rates, and temperatures during heat treatment cycles. The determination of hydrogen degassing anneals has chosen as the topic to test the use of a coupled external programme. This involved the calculation and summation of hydrogen degassing effects for all the heat treatment cycles experienced by an ingot, as discussed in section 5.3. Although DataEase allows the definition of global or temporary variables for storage of data values that can change during processing a procedure, it is cumbersome to deal with un-known numbers of variables and not as efficient as the programme written in C^{++}.

The evaluation that was carried out showed that such enhancement provides a promising method for dealing with complex manufactory problems. There is no need for a system additional to the DBMS, so that all the advantages of having a DBMS such as powerful data storage, concurrent access, and security are retained.

7.10 Assessment of the FEL CAPP System

As was reported in Chapter 6, the system described in this thesis is not currently operated by Forgemasters Engineering Ltd., the organisation for which, and in collaboration with which, it was developed. It is important to examine the extent to which this is due to fundamental errors in the conception and design of the system.

There were obvious operating difficulties which were not associated with the system design. The hardware platforms on which the system was developed were not appropriate for its day to day operation, particularly due to the absence of expanded memory and to their relatively slow speeds. Both of these features slowed the operation of the system very significantly, since data searches involved access to FEL's overall network system's server. The availability of expanded memory, and use of the DataEase update that allows the storage of procedures and data in expanded memory, would have resulted in significantly faster response times, as would the use, now, of faster PCs (e.g. 486, Pentium, etc.). It is unfortunate that concern for the speed of the system resulting from these hardware and software considerations clouded the potential users assessment about the conception and design of the system.

The assessment has also been clouded, to some extent, because the Metallurgical Engineers operating the system were unfamiliar with the use of PC based software and with keyboard entry of data into computer systems. These unfamiliarities aggravated difficulties associated with learning how to use newly designed data entry screens, and were further intensified when adequate time could not be allocated for operator training with the system when it was developed.

The speed of response of the system had an important influence on its assessment because the paper document system currently used at FEL has been developed over a large number of years, and is extremely efficient. A series of proformas has been designed and improved upon so that they are very easy to use. In part, this ease of use results from the extent to which manufacturing knowledge has been incorporated into the proforma design. This knowledge is manifest in the processing operations offered as field choices, the order in which these operations are presented on the form and the supplementary information that is requested. In effect, the manufacturing plan proforma, even before it is filled in, presents the user with a default manufacturing plan and requires decisions to be made in a logical sequence very close to the operating sequence. The manufacturing plan can then be designed merely by ticking boxes, sometimes inserting unusual operations into comment fields on the form. Although the computer system went some way to providing similar support for the design of manufacturing plans, the flexibility of the support offered by the paper based system could not be replicated completely.

The proformas were also designed to minimise the amount of data that had to be entered. Forging definition records, for example, did not have to be created by the process planners, the information that made up each forging identity being in existence on a number of forms, such as the order form, that had been completed prior to the process planning stage. Time spent in creating forging definition records added to the work load at the process planning stage, without the record creating any obvious benefit at the planning stage. The fact, for example, that forging definition records could provide information that would speed up the creation of test certification records did not compensate the process planners for the time they spent in creating the forging definition records in the FEL CAPP system. The wider extension of a computer system to incorporate sales records should avoid this type of problem.

Of course, the FEL CAPP system was designed as a replica of the paper based system, and this might be considered a major criticism of this work. Such criticism could be argued from the premise that a computer system should offer new data manipulation techniques that would allow it transcend the mode of operation of a paper based system. Whether or not this is true, there was no choice in this work other than to start by analysing the paper based system and then by replicating its capabilities in the computer system. This was because the paper based system represents the decision making practice actually required by FEL and the prime requirement of the computer based system was that it followed that practice.

The Quality Control procedures in place at FEL, and which represent a major component of the firm's manufacturing capital, were a further reason to replicate the paper based system rather than to replace it. The procedures require the creation and maintenance of certain documents at certain points in the manufacturing process. Obviously, the computer based system must create those documents if it is to comply with FEL's Quality Manual. In order to do this, the logic of the computer based system must lie very close to that of the paper based system, so the scope for the computer based system to be innovative is limited.

Given that the paper based system was in existence, had operated smoothly for a number of years and was enshrined in the FEL Quality Manual, an easier way to gain the confidence of the process planners in the use of computers might have been to present them with an 'intelligent form processing system'. Such a system would have assisted with preparation of documents for the paper based system, but would have relied on the paper based system for retention of documents once they had been completed and issued. Considerable system complication would have been avoided if this approach had been adopted since procedures to keep data secure, through the use of versional and facsimile files for example, would perhaps not have been required. It is very possible that such an approach might have won favour with the process planners and been accepted as no worse than, and possible an improvement over, the paper based system.

In this connection, it should be noted that the only part of the computer based CAPP system developed in this work still being operated at FEL is the part designed to control documents circulation in the paper based system. This related to the control of the issue of documents inside FEL and outside, to FSL and to customers. It is being operated, moreover, by staff with established keyboard skills who find the document automatic tracking facility conferred by the CAPP system database a very considerable advantage.

Although this suggests that designing the computer system as a 'handmaiden' to the paper based system might have provided an easier implementation route for computer usage in process planning at FEL, it would have ignored an important potential advantage of a computer based system - that of rapid data search through stored data in order to assess past performance. To be able to assess past performance is an increasingly important facility for senior management, in both the financial and quality fields. Invitations to tender for large manufacturing contracts frequently require a quality audit of relevant past manufacturing performance to be included in the tender. Considerable commercial advantage can thus be won by manufacturing firms able to conduct such audits rapidly and the existence of comprehensive computer records would obviously confer this ability. Thus a well designed computer process planning system, which included information about discrepancy reports etc., could assist a manufacturing firm, such as Forgemasters Engineering Ltd. in winning new orders. It would also assist in the continuous monitoring of the firm's manufacturing performance so that it could be a useful tool in the pursuit of greater manufacturing efficiency.

Neither would the development of an 'intelligent form processing system' for the process planners have raised the issues related to the creation of a computer based processing planning system for the manufacture of large steel forgings that have been explored in this work. It is unlikely that the need to designate separate planning areas for different aspects of the process planning task would have been seen so clearly. Nor would the need to use a versional approach been perceived and the value of facsimile files for intermediate planning purposes would not have been understood.

Thus a great deal has been learnt in this study about the issues that must be addressed when developing a process planning system for the 'bespoke' manufacture of large engineering components. Starting with the conceptual model of separate planning areas, versional files, facsimile files and the central identity file essential for linking purposes, a better computer based system could now be designed. Such a system would be less of a mimic of the paper based system, whilst retaining the documentation required in the Quality Manual, and could offer speed advantages over the current system. Once such a system had been designed in the light of the analysis carried out in this work, a more effective step by step implementation route could be devised that would ease acceptance of the system by those whom it was designed.

CHAPTER 8

CONCLUSIONS AND FUTURE RESEARCH

The research work reported in this thesis deals with two aspects of one approach to the automation of a process planning system:-

1. A relational database system which can integrate manufacturing functions into a single system allowing process planning, test result preparation and QA control activities to share data in a single relational database;

2. Enhancement of the database with external programmes through which manufacturing data are processed and engineering calculations are carried out to obtain solutions to the manufacturing problems presented.

After analysing the metallurgical data which Sheffield Forgemasters Engineering Ltd. use to construct the manufacturing plans for the various forgings they manufacture, the overall structure for a system to prepare multiple process plans has been established, which fills a gap in the current research on CAPP systems for manufacturing. Based on the structure, a relational database system has been created. the system has incorporated manufacturing knowledge and rules which can assist metallurgists in the design of process plans.

The operation of the computer aided process planning system closely relates to the model of the real-world information processes. The application of AI techniques in the FEL CAPP system has enabled the system to capture and utilise the expertise of process planners.

The following conclusions have been drawn through the establishment of the CAPP system for heavy steel forging environment:-

1) By incorporating metallurgical knowledge, the CAPP system can assist the metallurgical process planners to improve the quality of the process plans and sub-plans, and allow the plans to be developed more quickly.

2) Substantial safeguards have been developed to ensure that the engineering information remains secure and correct - safeguards such as the use of facsimile files and the versional approach.

3) Incorporating rules as a rule templates leads to a simple manufacturing rule structure which facilitates modifying, updating and adding to the rules.

4) It is easier to deal with complex manufacturing planning problems using database technology than by compiling human expertise into a rule based expert system.

Experience gained in the development of this project has shown that computer based CAPP systems will only become established in bespoke manufacturing activities, such as the manufacture of large steel forgings to order, if a clear and concerted strategy exists for their implementation. Although CAPP systems themselves must retain a high level of internal self consistency, they must be linked to other aspects of the overall data flow in the manufacturing entity - sales orders and quality assurance documentation etc. In planning such systems, the highly effective nature of existing paper systems and their established role in quality assurance procedures must be realised, and acknowledgement made than many of the overall improvements in efficiency will be made in activities such as report generation and control, performance monitoring, etc., rather than in the process planning function themselves. This has implications for manpower planning in connection with the development and implementation of computer based CAPP systems, and these must be addressed as part of the concerted implementation strategy that is required.

The work carried out in this research is far from being complete, it rather presented an analysis to the manufacturing problems in order to achieve a better computer based CAPP system. Although this research contributes to the domain of the computer aided process planning, there remains room for improvement. The following is a list of a few considerations to achieve a better CAPP system for the process planning environment of heavy steel forging:-

1. Wider extension of a computer system should be included to incorporate sales records. This can be achieved by importing sales records from sales database to the database used for FEL CAPP system, and downstream to the preparation of test reports, etc.

2. Consideration such as assessment of past performance should be added to the FEL CAPP system, so that firm's manufacturing performance can be continually maintained.

3. The test results function should be extended to allow the evaluation of past process plans to inform current process plan design. For example, it may be possible to design programmes to carry out this evaluation task routinely and store the results in other files. When a new process plan design is started, the information stored in these files can be surveyed to alert the user with any past difficulties.

4. Graphic facility should be incorporated into the system to allow users to sketch forging design, and to prepare visual examination reports. The external enhancement approach could be used to achieve this application.

5. The rule criteria currently incorporated in the procedures for the steel melting range design can sometimes give more than one optimal range. It would be possible to select a single best range from amongst these if more rules are incorporated into the design process. Cost factors should be included to select the cost effective melting range.

In addition to these extensions to the system developed in this work, the provision of up-to-date software and appropriate hardware platforms are seen to be essential for the successful implementation of an effective CAPP system.

LIST OF REFERENCES

[AB85] James S. Albus, Anthony J. Barbera and M.L.Fitzgerald, "A Control System for An Automated Manufacturing Research Facility", pp. 28-44, 1985.

[AC93] Abdou, G.; Cheng, R., "TVCAPP, tolerance verification in computer-aided process planning", International Journal of Production Research v. 31, n. 2, Feb. 1993. pp. 393-411.

[AG87] AI-Zobaidie A and Grimson J B, "Expert systems and Database systems: how can they serve each other?", Expert Systems, Vol 4 No. 1 (February), pp. 20-37, 1987.

[AM89] Lisa A Abraham and Steve Mohn, "Expert Database Systems in Manufacturing", Process Advances in Instrumentation, Vol. 44, PT. 2, 1989.

[AT88] Abarbanel R, Tou F, and Gilbert V, "A Bridge between Database and Knowledge Bases" in AI Tools and Techniques, M. Yazdani and M. Richer, Eds. Ablex. Norwood, 1988.

[AZ89] Leo Alting and Hongchao Zhang, "Computer Aided Process Planning: the state-of-the-art survey", Int. Journal of Prod. Res., Vol. 27, No. 4, pp. 553-585, 1989.

[B86] Hamid R. Berenji, "Use of Artificial Intelligence in Automated Process Planning", Computer in Mechanical Engineering, Sep. 1986/47.

[B86i] W.J. Black, "Intelligent Knowledge Based Systems", Van Nostrand Reinhold (UK) Co. Ltd, 1986.

[B90] Nikola bogunovic, "Coupling an Expert System with Industrial Real-Time Processes", IECON PROC. I.E.E.E., Vol. 2, 1990.

[B90i] Bhaskaran, K., Process Plan Selection, International Journal of Production Research. 28 (8), pp. 1527-1539. 1990.

[B91] Paul Beynon-Davies, "Expert Database Systems - A Gentile Introduction", McGraw-Hill Book Company, 1991.

[BB92] Eric A Brandes and G.B. Brook, Smithells Metals Reference Book, Butterworth-Heinemann Ltd., Sixth Edition, London, 1992.

[BD90] W. Bettendorf and R.H.Davis, "Knowledge-based Support for Process Planning in a Forging Environment", Appl. Math. Modelling, Vol. 14, Mar., 1990.

[BH87] S.L.Brooks, K.E.Hummel and M.L.Wolf, "XCUT: A rule-based Expert System for the Automated Process Planning of Machined Parts", IBID, pp. 53-56. 1987.

[BG89] Bodkin T and Grahm I, "Case Studies of Expert Systems Development Using Microcomputer Software Packages", Expert Systems, Vol. 6, No. 1, pp. 12-16., 1989

[BM83] Brodie M., Mylopoulos J, Schmidt J., Conceptual Modelling: Perspectives from Artificial Intelligence, Databases and Programming Languages, Springer-Verlag, 1983.

[BM84] Brodie M.L, Mylopoulos J, Schmidt J.W, On Conceptual Modelling: Perspectives from Artificial Intelligence, Databases and Programming Languages, Springer Verlag, New York, 1984.

[BM86] Michael L Brodie and John Mylopoulos, "Topics in Information System - On knowledge Base Management systems", 'Integrating Artificial Intelligence and database technologies, Springer-Verlag New York, 1986.

[C79] Codd E F., "Extending the Database Relational Model to Capture More Meaning", ACM Trans. Database Syst. 4, 4 (Dec. 1979), 3397-434

[C87] CRYSTAL, Intelligent Environments, PC-DOS International Business Machine Corp.; MS-DOS Microsoft "C" - Microsoft Corporation; Lotus 123, Symphony - Lotus Development Corp. etc. Copyright (c) 1987.

[C89] Chakravarthy S., "Rule Management and Evaluation: An Active DBMS Perspective", Sigmod Record, Vol. 18, No. 3, Sep. 1989.

[CM89] Cheiney J and C. Maaindereveille, " Relational Storage and Efficient Retrieval of Rules in a Deductive DBMS", Data Engineering Conference, IEEE, 1989.

[CT87] M. R. Cutkosky and J. M. Tenenbaum, "CAD/CAM Integration through Concurrent process and Product Design", Intelligence and Integrated Manufacturing Analysis and Synthesis. ASME, 1-10, 1987.

[CW84] Tien-Chien Chang and Richard A. Wysk, "Integrating CAD and CAM through automated Process Planning", Int. J. prod. Vol. 22, No. 5, pp. 877-894, 1984.

[CS92] Computer Shopper, Number 53, July, 1992 and Number 63, Feb. 1993..

[CW85] Chang T. C. and Wysk R. A. "An Introduction to Computer-Aided Process Planning Systems", Prentice Hall-Publishing Company.

[CY91] Chiang P. T. and Yang J. C. S., "Development of a process determination and sequencing technique for a manufacturing system", International Journal of Production Research, 29 (3), pp. 513-535, 1991.

[D85] Debenham J, "Knowledge Base Design", Australian Comput. J. Vol 17 No. 1 (February) pp. 42-48, 1985.

[D86] Date, C.J., Relational Database: Selected Writings. Addison-Wesley. Reading, Mass., 1986.

[D87] Dell K. Allen, "An Introduction to Computer-Aided Process Planning", Computer-Aided Process Planning, pp. 7-21, Fall 1987.

[D90] Date, C.J., An Introduction to Database System, Addison-Wesley Publishing Co. Vol. 1, Fourth Edition, 1990.

[D91] Paul Beynon-Davies, Expert Database Systems - A Gentle Introduction, McGraw-Hill Book Company, 1991.

[DB88] Dayal U., Buchmann A. P. and McCarthy D. R., "Rules Are Objects Too: A Knowledge Model for an Active, Objective-Oriented Database System", Proceedings of 2nd International Workshop on Object-Oriented Database Systems, Springer-Verlag, New York, pp. 129-143, Sep. 1988.

[DD84] B. J. Davies and I. L. Derbyshire, "The Use of Expert System in Process-Planning", Annals of the CIRP, Vol. 33/1, pp. 303-306, 1984.

[DG88] Edward R Dougherty and Charles R Giardiana, "Mathematical Methods for Artificial Intelligence and Autonomous Systems", Prentice-Hall International, Inc. 1988.

[DI91] DataEase International, Reference Manuals: User's Guide, Query Language Guide, Administration and Developer's Guide. 1990-1991.

[DL85] Descotte, Y. and Latombe, J-C. Making Compromises Among Antagonist Constraints in a Planner", Artificial Intelligence, vol. 27, pp. 183-217, 1985.

[F82] C. Forgy, "A Fast Algorithm for the Many Pattern - any Object Pattern Match Problem", Artificial Intelligence, Vol. 19, 1982.

[F87] Florentin J J, "KEE", Expert Systems, 4(2), pp. 118-247, 1987.

[FC93] Jerry Y. H. Fuh, Chao-Hwa Chang and Michel A. Melkanoff, "An Integrated Fixture Planning and Analysis system for Machining Processes", Robotics & Computer-Integrated Manufacturing, Vol. 10, No. 5, pp. 339-353, 1993.

[G88] Gunther Kruse, "Excellence in Manufacturing", Production Engineer, April, 1988, pp.54-56.

[G92] Greg Perry, C++ By Example, Programming Series, Ver. 1.0, 1992 by Que.

[GL87] B.K. Gustafson and S.C.Y. Lu, "Fundamental Issues in Applying Artificial Intelligence to Industrial Automation: A Case Study of Automated robot task Programming", American Society of Mechanical Engineers, Publ. by ASME, New York, USA. pp. 365-380, 1987.

[GL88] Gray P.M.D. and R.J. Lucas, Proglog and Database - Implementations and New Directions, Ellis Horwood Limited, England, 1988.

[GN92] Germond A J and Niebur D, "Survey of Knowledge-Based Systems in Power Systems: Europe", Proceedings of the IEEE, Vol. 80, No. 5, May 1992.

[H86] K.J. Hammond, CHEF: a model of case-based planning. Proc. 5th National Conf. on Artificial Intelligence - 1986, Menlo Park, CA, USA, Vol. 1, pp. 267-271.

[H90] I.T. Hawryszkiewycz, "Relational database design", by Prentice Hall of Australia Pty Ltd, 1990.

[HS91] B. Humm, Ch. Schulz, M. Rastke and G. Warneche, "A System for Case-based Process Planning", Computer in Industry, No. 17, pp. 169-180, 1991.

[H92] Shane Y. Hong, "Self-learning System for Knowledge Based Diagnoses of Drill Condition in Full Automatic Manufacturing System", SPIE Vol. 1707, Applications of Artificial Intelligence X: Knowledge-Based Systems, 1992.

[HH91] David P. Hale, Joanne E. Hurd, and George M. Kasper, " A Knowledge Exchange Architecture for Collaborative Human-Computer Communication", IEEE Transactions on Systems, Man, and Vybernetics, Vol. 21, No. 3, May/June, 1991.

[HK85] Paul Harmon, David King, J, Wiley, "Expert system: Artificial Intelligence in Business", 1985.

[HL88] Inyong Ham and Stephen C.-Y. Lu, "Computer-Aided Process Planning: The Present and the Future", Annals of the CIRP Vol. 37/2/1988.

[HM78] Hammer M and Mcleod D, "The Semantic Data Model: a modelling mechanism for database applications. In Proceedings of the 1978 ACM-SIGMOD Conference (Austin. Tex. May 31 - June 2), ACM, New York, 1978, pp. 26-35.

[HW91] Tung-Hsu (Tony) Hou and Hsu-Pin (Ben) Wang, "Integration of A CAPP System and An FME", Computers Ind. Eng., Vol. 20, No. 2, pp. 231-242, 1991.

[II86] Ishikawa H, Izumida Y and Yoshino T et. "A knowledge-based approach to design on a portable natural language interface to database systems" IEEE Proc. Conf. on Data Eng. Los. Angeles, USA pp. 134-143, 1986.

[IK87] M. Inui, A. Kinosada, F. Kimura and T. Sata, "Automatic Process Planning for Sheet Metal Parts with Bending Simulation", American Society of Mechanical Engineers, Publ. by ASME, New York, USA. pp. 245-258, 1987.

[J88] Glenn A. Jackson, "Relational Database Design with Microcomputer Applications", Prentice-Hall International Editions, 1988.

[JC89] Bob Jansen and Paul Compton, "Data dictionary Approach to the Maintenance of Expert Sytems: The Knowledge Dictionary", Knowledge-Based Systems, Vol 2, No. 1, March, 1989.

[JF89] Joseph Fiksel and Frederick Hayes-Roth, "Knowledge systems for Planning Support", IEEE Expert, Fall 1989,

[JR87] S. Jablonski, T. Ruf, H. Wedekind, G. Zoerntlein, "A Data Distribution Mechanism for Manufacturing Applications", IEEE, 1987.

[JS90] Juri, A. H., Saia, A., and Pennington, A., "Reasoning about maching operations using feature-based models", International Journal of Production Research, 28 (1), pp. 153-171, 1990.

[JV84a] Jarke M and Vassiliou, "Coupling expert systems with database management systems", In Reitman, 1984.

[JV84b] Jarke M and Vassiliou, "Databases and expert systems: opportunities and architecture for integration" In Gardarin and Gellenbe, 1984.

[JV88] Joshi, S., Vissa, N. N., and Chang, T. C., "Expert Process Planning System with Solid Model Interface", Internaltional Journal of Production Research, 26 (5), pp. 863-885, 1988.

[K86] Kerschberg L., Ed. Expert Database Systems, Benjamin Cummings, Menlo Park, Calif. 1986.

[K90] Peter Koch, "PC-range Expert System Shells: Advantages and Problems", Fuzzy Sets and Systems, Vol. 41, No. 2, pp. 127-131, May, 1990.

[KB92] Korde U P, Bora B C, Stelson K A and Riley D R, "Computer Aided Process Planning for Turned Parts Using Fundamental and Heuristic Principles" Journal of Engineering for Industry, Vol 114, n 1, pp. 31-40, Feb. 1992.

[KL84] Randy H. Katz and Tobin J. Lehman, "Database Support for versions and Alternatives of Large Design Files", IEEE Transactions on Software Engineering, vol. Sec-10, No. 2, March, 1984.

[KH88] Soundar Kumara and Inyong Ham, "Database Considerations in Manufacturing Systems Integration", Robotics & Computer-Integrated Manufacturing, Vol. 4, No. 3/4, pp. 571-583, 1988.

[L76] Link, C.H., "CAPP-CAM-I Automated Process Planning System", Proceedings, 13th Numerical Control Society Annual Technical Meeting and Conference, Cincinnati.

[L86] Lu S.C-Y., "Knowledge-Based Expert Systems: A New Horizon of Manufacturing Automation", Winter annual meeting technical papers, ASME Production Engineering Division, Dec. 1986, Anaheim, CA.

[L91] Robert Lafore, "Object-Oriented Programming in Turbo C++", First Edition, 1991, by The Waite Group, Inc.

[LL90] Lee, I. B. H., Lim, B. S., and Nee, A. Y. C., "Object-oriented rule based expert process planning system for the manufacture of progressive dies", Proceedings of Industrial Automation (IA) '90 Conferences, Singapore, pp. 393-412, 1990.

[LL93] Lee, I. B. H., Lim, B. S., and Nee, A. Y. C., "Knowledge-based process planning system for the manufacture of progressive dies", INT. J. PROD. RES., Vol. 31, No. 2, pp. 251-278, 1993.

[M82] Matsushima, K., et al. "The integration of CAD and CAM by application of artificial intelligence techniques", Annals of the CIRP, 31 (1), 1982.

[M87] Peter Marks, "Expert System-Based Generative Process Planning", CASA/SMS Autofact'87 Conference, November 1987.

[M91] Sitansu S. Mittra, "Principles of Relational Database Systems", Prentice-Hall International Editions, 1991.

[MW86] Missikoff M and Wiederhold G, "Towards a unified approach for expert and database systems", In Kerschberg, 1986.

[N87] Gerhard Neipp, "Artificial Intelligence and its Impact on Industry: A New dimension within the Framework of Computer integrated Manufacturing (CIM), Int. J. Technology Management, Vol. 2, Nos 5/6, 1987.

[NC85] Nau, D. S. and Chang, T. C. "A Knowledge-based Approach to Generative Process Planning", Computer-Aided/Intelligent Process Planning (USA: ASME), pp.65-71, 1985.

[NH88] Napheys B and D Herkimer, "A Look at Loosely Coupled Prolog/Database Systems", Proceedings from the Second International Conference on Expert Database Systems, The Benjamin/Cummings Publishing Company, Redwood City, Cal. USA, 1988.

[P91] M. J. Page, "Capturing Manufacturing Experiece Using Computer Aided Process Planning", GEC Review, Vol, 7, No. 1, 1991.

[PP91] Veena Paul and Ramakrishna V. Polamraju, "Rule Management and Inferenceing in Relational Databases", Conference Proceedings of Southeastcon, Vol. 2, IEEE CAT No. 91CH2998-3, pp. 695-697, 1991.

[R90] Bharat Rupaarel, "Designing and Implementing Intelligent Database Applications - A Case Study", Programs and Project Proc 90 IEEE Conference, Manage Expert Systems Program Project, Published by IEEE, 1990 Sep. 10-12, pp. 223-229.

[R90i] Ricardo Catherine M, Principles, Design, and Implementation, New York, London, Collier Macmillan, 1990.

[RB94] Roy A, Burns G and Champenari R, "Feature Extraction Using AI_Technique for Component Identification and Analysis", Applications of Artificial Intelligence in Engineering, pp. 141-148, 1994.

[RD90] Richard G Ramirez, Ronald Dattero and Joobin Choobineh, "Representing Generalisations and Exceptions in Expert Database Systems", Decision Support Systems, 6 (1990) pp. 26-44.

[RI92] J. Richard, Z. Idelmerfaa, F. Lepage, and M. Veron, "Quality Control in a Flexible Manufacturing Cell", Annals of the CIPP, Vol 41/1, pp. 561-564, 1992.

[RM91] Nick Roussopoulos, Leo Mark, Timos Sellis, and Christos Faloutsos, "An Architecture for High Performance Engineering Information Systems", IEEE Trans. Software Eng., Vol. 17, No. 1, Jan. 1991.

[RR86] Risch T, and Reboh R, Syntel: Knowledge Programming Using Functional Representations. In Proceedings of AI-86 Morgan Kaufman, Los Altos, Calif., 986, pp. 1003-1007.

[RR88] Tore Risch, Rene Reboh, Peter Hart, and Richard Duda, "A Functional Approach to Integrating Database and Expert Systems", Communications of the ACM, Dec. 19988, Vol 31, No. 12, pp. 14241437.

[RW87] Lawrence A. Rowe and Christopher B. Williams, "An Object-Oriented Database Design for Integrated Circuit Fabrication", IEEE, 1987.

[S52] Reinhardt Schumann, JR., "Metallurgical Engineering", Engineering Principles, Vol. 1, Addison-Wesley Publishing Company, Inc. London 1952.

[S62] Smialowski Michael, Hydrogen in Steel, Effect of Hydrogen on Iron and Steel during Production, Fabricating and Use, Oxford, Pergamon, 1962.

[S63] Shewen Paul G, Diffusion in Solids, New York, London, McGraw-Hill, McGrw-Hill Series in Materials Science and Engineering, 1963.

[S80] Schaffer, G. "GT vs Automated Process Planning", American Machinist, pp. 119-122, May, 1980.

[S82] R. Schank, Dynamic Memory: A Theory of Reminding and Learning in Computers and People. Canbridge University Press, Cambridge, 1982.

[S86] Szolovits P, "Knowledge-Based Systems: A Survey", in Preceding of on knowledge base management systems, Springer-Verlag, pp. 339-352, 1986.

[S86i] Stonebraker M, Triggers and Inference in Database Systems. In On Knowledge Base Management Systems. M. L. Brodie and J. Mylopoulos, Eds. Springer-Verlag, New York, 1986, pp. 297-314.

[S91] Conor Sexton, "C Programming - Pocket Book", First Published in 1991, "C++ Programming - Pocket Book", First published in 1993, Newnes, A Butterwoth-Heinemann Book.

[SB88] Alojz Sluga, Peter Butala, Nada Lavrac and Matjaz Gams, "An Attempt to Implement Expert System Techniques in CAPP", Robotics & Computer-Integrated Manufacturing, Vol. 4, No. 1/2, pp. 77-82, 1988.

[SE82] P.J. Sackett and S. Evans. Implementing Computer Aided Process Planning. International Journal of Operation & Production Management 1982; Vol. 2 No. 2, pp. 21-29.

[SH82] J.A. Simpson, R.J.Hocken, and J.S.Albus, "The Automated Manufacturing Research Facility of the National Bureau of Standards", Journal of Manufacturing Systems, Vol. 1, No. 1, pp. 17-31, 1982.

[SH87] F. J. Smith and J G Hughes, "Some Special Features of Material Database System", IEEE 1987.

[SL76] D. G. Severence and G. M. Lohman, "Differential files: Their application to the maintenance of large databases", ACM Trans. Database System, Vol. 1, Sep. 1976.

[SR92] Rajiv Singh and Shivakumar Raman, "METEX-An Expert System for Machining Planning", INT. J. Prod., Vol. 30, No. 7, pp. 1501-1516, 1992.

[T86] Joseph Tulkoff, Computer Aided process Planning, Published by Computer and Automated System Association of SME, 1985.

[T89] Andrew Taylor, "An Investigation of Knowledge-based Models for automated Procedure Generation in the Arc Welding Domain", Int. J. Prod. Res., 1989, vol. 27, 97 No. 11, 1855-1862.

[TK88a] Costa Tsatsoulis and R.L.Kashyap, "A System for Knowledge-based Process Planning", Artificial Intelligence in Engineering, Vol. 3, No. 2, pp 61-75, 1988,

[TK88b] Costas Tsatsoulis and R.L.Kashyap, "A Case-Based System for Process Planning", Robotics & Computer Manufacturing, Vol. 4, No. 3/4. pp. 557-570, 1988.

[VK86] P. Valduriez, S. Khoshafian, and G. Copeland, "Implementation Techniques for Complex Objects", Proc. 12th Int. Conf., Very large Data Bases, Tokyo, Japan, Aug. 1986.

[W84] Wiederhold G, "Knowledge and Database Management", IEEE Software, 1, 1, 1984, pp. 63-73.

[W91] Wang, Ming-Tzong, "Object-oriented feature-based CAD/CAPP/CAM integration framework", Advances in Design Automation American Society of Mechanical Engineers, Design Engineering Division (Publication) DE v. 32, pt 1. Publ by ASME, New York, NY, USA. p 109-116, 1991.

[W92] Wang, Eric, "Using automatic feature recognition to interface CAD to CAPP", Computers in Engineering - 1992. Publ by ASME, New York, NY, USA. pp. 215-231,1992.

[W93] Hans-Peter Wiendahl, "Production Planning and Control - The Tool to Ensure Logistical Quality", Robotics & Computer-Integrated Manufacturing, Vol. 10, No. 1/2, pp. 99-107, 1993.

[WW88] Hsu-Pin Wang and Pichard A. Wysk, "A Knowledge-Based Approach for Automated Process Planning", INT. J. Prod. Res. Vol. 26, No. 6, pp. 999-1014, 1988.

[YS87] K.M. Yin and D. Solomon, "Using Turbo Prolog, Que Corporation, Indianapolis, 1987.

[Z86] Zimmers E. W., "Using simulation for process planning", CIM Review, 3 (1), pp. 18-24, 1986.

APPENDIX 1

Fields, Relations and Their Definitions

1-1 Fields in the Various Status Files

1-1-1 Forging Status

Description: This relation is the only file that can not normally be accessed by system users. It is maintained by system procedures and is used to identify Forging Definition key fields, ID_No. and ID_Ver., for each unique forging order number, FELOrder No.

1. ID_No. — Num.String of 5, the first identifier in the Forging Definition relation.
2. ID_Ver. — Text of 1, Issue Letter. The second identifier in THE Forging Definition relation. (ID_No. and ID_Ver. are used as the composite primary key in the Forging Definition relation.)
3. FEL Order No. — Num.String of 7. (The unique number used to eliminate the redundant job to be entered in the database)

1-1-2 MP_Status (Manufacturing Plan Status)

Description: This relation is used to link Manufacturing Plan with Forging Definition file in order to draw information stored in Forging Definition file. It also shows all related FEL Order Numbers to which the Manufacturing Plan is dedicated.

1. MP_No. — Text of 5, the first identifier in Manufacturing Plan.
2. MP_I — Text of 1, Issue Letter, the second identifier in Manufacturing Plan relation. (MP_No. and MP_I are used as the composite primary key in the Manufacturing Plan relation.)
3. Works_O_No. — Num.String of 11, it is a foreign key from the Forging Definition for works order No.(s).
4. ID_Ver. — Text of 1, the second identifier in Forging Definition. (The composite key of Works_O_No. and ID_Ver. are used as the alternative key to identify an unique record in Forging Definition relation.)
5. FIP? — Choice of Yes or No, to indicate whether or not the forgings are in progress in the Manuf. Plan.

1-1-3 QP_Status (Quality Plan Status)

Description: This relation is used to link Quality Plan with Forging Definition file in order to draw information stored in Forging Definition file. It also shows all related FEL Order Numbers to which the Quality Plan is dedicated.

1. QP_No. — Text of 5, the first identifier it the Quality Plan relation.
2. QP_I — Text of 1, Issue Letter, the second identifier in Quality Plan relation. (QP_No. and QP_I are used as the composite primary key in Quality Plan relation.)
3. Works_O_No. — Num.String of 11, it is a foreign key from Forging Definition for works order No.s)
4. ID_Ver. — Text of 1, the second identifier in Forging Definition. (The composite key of Works_O_No. and ID_Ver. is used as the alternative key to identify an unique record in Forging Definition.)
5. FIP? — Choice of Yes/No, to indicate whether or not the forgings are in progress in the Quality Plan.

1-1-4 SO_status (Steel Order Status)

Description: This relation is used to link Steel Order with Forging Definition file through Forging Status file in order to draw information stored in Forging Definition file. It also shows all related FEL Order Numbers to which the Steel Order is related.

1. ST_No. Num.String 6, the first identifier in the Steelord.
2. Version Text 1, the second identifier in the Steelord.
   (The composite key of ST_No. and Version is used as the primary key in Steelord relation.)
3. FEL Order No. Num.String 7, to show how FEL Order No.s are related to the Steel Orders. It is a foreign key from the Forging Status relation and used to link them together.)
4. FIP? Choice of Yes or No to indicate whether or not the forgings are in progress in this Steel Order.
5. Item_No. Num.String 5, the first identifier in the Steelmaking programme relations.
6. Item_Issue Text of 2, the second identifier in Steelmaking Programme relations.
   (Item_No. and Item_Issue are used as the composite primary key in the Steelmaking Programme relations.)
7. Cast_No. Text of 5, the first identifier in the Cast Analysis relations.
8. Cast_Letter Text of 5, the second identifier in the Cast Analysis relations.
   (Cast_No. and Cast_Letter are used as the composite primary key in the Steelmaking Programme relations.)

1-2 Fields in the Relations in the Forging Definition Planning Area

1-2-1 Forging Definition

Description: Forging Definition file stores common data which can be shared by other relations within the database. It identifies, for each bundle of FEL Order Numbers, the customer, the specification requirements, the record creation date the order and dispatch date, etc.

1. ID_No. Num.String of 5, the first identifier in this relation. The value will never be changed
   for all its versions.
2. ID_Ver. Text of 1, Issue Letter. the second identifier.
   (ID_No. and ID_Ver. are used as the composite primary key in the Forging Definition relation.)
3. Works_O_No. Num.String 11, to indicate a certain range of FEL Order No.(s) related.
   (Both Works_O_No. and ID_Ver. can be used as the alternative key to identify a unique record when the Works Order Number is preferable.)
4. RP? Text of 2, to indicate whether the forgings are replacement or not.
5. Date_Crd Date of the record creation.
6. Initial Text of 15, for the initiator's name.
7. Customer_No. Text of 8, to identify the customer in the Customer relation.
8. Sales_No. Num.String 8, indicating sales reference number.
9. FES Num.String 5, the first identifier in the Specification relation.
10. Group Text of 1, the second identifier in the Specification relation.
    (FES and Group are the composite primary key in the Specification relation.)
11. Description Text of 42, to indicate what the forgings are.
12. Application Text of 27, to indicate the forging's application.
13. Product Cat's Choice of product category.
14. Cust. Drawing Text of 31, to indicate the drawing numbers or dimensions.

15. FEL Draw. No.	Text of 31, to indicate the FEL drawing numbers.
16. Customer Order	Text of 35, to show the customer order number.
17. Inspection	Text of 39, to show the specified inspection authority.
18. Other Spec.	Text of 56, to show any specifications, other than that defined by fields 9 and 10.
19. Comment	Text of 68, for comments.
20. Date_Ordered	Date, for ordered date.
21. Date_Delivery	Date, for delivery date.

1-2-2 Specifications

Description: This relation is used to link a chemical range with several set of mechanical properties. It can also indicate, for each chemical range under certain conditions, what the mechanical properties will be.

1. FES	Num.String 5, the first identifier in this relation.
2. Group	Text of 1, the second identifier in this relation. (FES and Group are the composite primary key.)
3. Approve	Choice of Y/N, to indicate whether or not the Spec. has been approved.
4. Initial	Text of 15, name of the person who creates the Spec.
5. Date	Date, that the Spec. is created.
6. Spec. Cat's	Choice of Specification categories, to show whether it is a customer Spec., FEL Spec., or international standard Spec., etc.
7. Cond._1	Text of 8, to indicate heat treatment cycle.
8. Cond._2	Text of 8, to indicate forging size or others.
9. Comment	Text of 68, any comments on the Specification.

1-2-3 Chemical Range

Description: This relation is used to store chemical range for up to 22 elements with Max. and Min. values. Fields 2-5 indicate precisely the meaning of the FES identifier.

1. FES	Num.String 5, the unique Spec. code for the sake of brevity to represent the composite attributes {Spec. Source, Spec. Name, Grade, Issue}.
2. Spec. Source	Text of 10, to indicate the Spec. sources.
3. Spec. Name	Text of 12, to indicate the Spec. names.
4. Grade	Text of 15, to indicate the Spec. grades.
5. Issue	Text of 15, to indicate the Spec. issues.
6. CMin	Num.String, minimum content of Carbon.
7. CMax	Num.String, maximum content of Carbon.
8. SiMin	Num.String, minimum content of Silicon.
9. SiMax	Num.String, maximum content of Silicon.
10. MnMin	Num.String, minimum content of Manganese.
11. MnMax	Num.String, maximum content of Manganese.
12. PMin	Num.String, minimum content of Phosphorus.
13. PMax	Num.String, maximum content of Phosphorous.
14. SMin	Num.String, minimum content of Sulphur.
15. SMax	Num.String, maximum content of Sulphur.
16. CriMin	Num.String, minimum content of Chromium.
17. CrMax	Num.String, maximum content of Chromium.
18. Other elements	(There are 22 other solute elements with Min. and Max. value fields, similar to Carbon or Silicon. 18 elements are named, four are available for definition by the user.)

1-2-4 Mechanical Property

Description: This relation records mechanical properties for each chemical range identified by FES number. It indicates mechanical properties on different directions and orientations.

1. FES	Mum.String 5, the first identifier in this relation.
2. Group	Text of 1, the second identifier in this relation. (Both FES and Group are the composite primary key in the Specification relation and used to group the mechanical properties for the same FES and Group.)
3. No.	Mechanical Property number. the third identifier. (the composite of FES, Group and No. can be used to identify each unique mechanical property record.)
4. Dir.	Choice of directions, e.g. axial, radial etc. to show the direction in which the test sample should be taken.
5. Location	Text of 6, to show the location of the test sample.
6. Section	Text of 6, to show the section from which the test sample will be taken.
7. Ten_Temp.	Text of 4, to show tensile temperature.
8. Ten_Unit	Choice of units in which tensile test to be recorded.
9. Ten_Value 1	Text of 5, to show the minimum tensile test value.
10. Ten_Value 2	Text of 5, to show the maximum tensile test value.
11. RM1	Text of 5, to show the minimum RM value.
12. RM2	Text of 5, to show the maximum RM value.
13. A.	Text of 3, showing minimum percentage elongation after fracture.
14. Z.	Text of 3, showing minimum reduction in area.
15. Imp_Temp.	Text of 4, to show the impact test temperature.
16. Imp_Unit	Choice of test units in which impact test to be recorded.
17. Imp_Value 1	Text of 5, to show the impact test value.
18. Imp_Value 2	Text of 5, to show the reference impact test value.
19. Bend_Angle	Text of 3, to indicate the bend test angles.
20. Bend_Rad.	Text of 3, to indicate the bend test radius.
21. Hardness 1	Text of 4, to show the minimum hardness test value.
22. Hardness 2	Text of 4, to show the maximum hardness test value.
23. Shear	Text of 3, to show the shear percentage value.
24. FATT	Text of 4, to show the FATT value.
25. Remarks	Text of 28, for comment on each mechanical property.

1-2-5 Customer

Description: This relation records customer information such as addresses contact telephone number and etc. Each record is identified by Customer_No..

1. Customer_No.	Text of 8, the unique number to identify each customer.
2. Customer Name	Text of 40, to indicate the customer name.
3. Address_1	Text of 20, for the first line of the address.
4. Address_2	Text of 20, for the second line of the address.
5. Address_3	Text of 20, for the third line of the address.
6. Post_Code	Text of 10, for the post code.
7. Tel. No.	Text of 12, for the telephone number.
8. Fax. No.	Text of 12, for the Fax. number.

1-3 Fields in the Relations in the Manufacturing Plan Planning Area

1-3-1 Manufacturing Plan

Description:	This relation is designed to record basic information in Manufacturing Plans. It is used as the main form which can be used to draw data from other relations. Other relations may have several records related to the record stored in this relation and these records are normally shown as sub-forms.
1. MP_No.	Text of 5, the first identifier in this relation.
2. MP_I	Text of 1, Issue Letter, the second identifier. (MP_No. and MP_I are used as the composite primary key in this relation.)
3. QP?	Choice of Yes/No, to indicate whether or not a related Quality Plan exists.
4. Date_Crd	Date, to indicate the date that the Manuf. Plan is created.
5. Date_edi	Date, to indicate the date that the Manuf. Plan is edited.
6. Initial	Text of 15, to indicate the person who has created it.
7. RMC	Num.String of 7, the primary key in RMC_Scrap relation.
8. Modify	Choice Yes/No, to show whether the RMC code has been modified.
9. Steelmaking1	Comment 1 for steelmaking requirements.
10. Steelmaking2	Comment 2 for steelmaking requirements.
11. Block	Choice of Yes/No, to indicate whether U/S block is applied.
12. Position1	Text 62, to indicate the test position requirements.
13. Position2	Test 62, to indicate the test position requirements.
14. Position3	Test 62, to indicate the test position requirements.
15. Sketch	Choice of Yes/No, to indicate if a sketch is attached.
16. Assur1	Choice field, to indicate the QA. standard.
17. Assur2	Choice field, to indicate the QA. standard.
18. Others1	Text field, to indicate other QA. standard.
19. Others2	Text field, to indicate other QA. standard.
20. Comment1	Text field, for the general comment.
21. Comment2	Text field, for the general comment.
22. Comment3	Text field, for the general comment.
23. Comment4	Text field, for the general comment.
24. Comment5	Text field, for the general comment.

1-3-2 MP_Status

(Already listed in Appendix 1.2)

1-3-3 Operations

Description:	This relation is designed as the main sub-form for Manufacturing Plan. It records operational sequences, comments to each operation, QA control codes and production time.
1. MP_No.	Text of 5, the first identifier in the Manufacturing Plan relation.
2. MP_I	Text of 1, Issue Letter, the second identifier in the Manufacturing Plan relation. (MP_No. and MP_I are used to group all the operations in one Manufacturing Plan.)
3. Seq.	Number 2, for the line sequence number of operations.
4. Op_No.	Number 2, for the operation number.
5. Operation	Choice of operations.
6. Comment	Text 41, for comment on each operation.
7. Act	Text 3, to record QA control codes.
8. Prod_Time	Num.String 4, to indicate production schedule time.

1-3-4 Test_Units

Description:	This relation is designed to record manufacturing plan test requirements. Several test requirements can be recorded in this relation and shown as a sub-form in Manufacturing Plan.
1. MP_No.	Text of 5, the first identifier in the Manufacturing Plan relation
2. MP_I	Text of 1, Issue Letter, the second identifier in the Manufacturing Plan relation. (MP_No. and MP_I are used to group all the tests in one Manuf. Plan for the same MP_No. and MP_I.)
3. Test_No.	Number, to indicate the test number. (The composite of MP_No., MP_I and Test_No. can be used to identify each unique test required in the Manufacturing Plan.)
4. Tests	Choice of tests to be undertaken,
5. Unit	Choice of units for each test.
6. Test_Com.	Text of 40, for each test comment.

1-3-5 RMC_Scrap

Description:	This relation records RMC (Raw Material Control) scrap codes. Each manufacturing plan needs to record RMC code to draw information in its relation for handling turning and scrapping of forgings.
1. RMC	Num.String 7, Primary key for each unique RMC record.
2. Scrap_Code	Text of 4, to indicate the scrap code.
3. Turnings	Text of 4, to indicate the turnings code.
4. Comment	Text of 35, for any comment on the RMC.

1-4 Fields in the Relations in the Quality Plan planning Area

1-4-1 Quality Plans

1-4-1-1 Quality Plan

Description:	This relation is designed to record basic information in quality plan. It is used as the main form which can be used to draw data from other relations. Other relations may have several records related to the record stored in this relation and these records are normally shown as sub-forms.
1. QP_No.	Text of 5, the first identifier in this relation. (It has the same value as the MP_No. to which the FEL Order Numbers are applied.)
2. QP_I	Text of 1, the second identifier in this relation. (the primary key is the composite field of QP_No. and QP_I to identify an unique record.)
3. MP_I	The Manuf. Plan Issue at the time the Quality Plan was issued. (used to trace the historical issues of the Manuf. Plan.)
4. Date_Edi	Date, to indicate the latest editing date.
5. Initial	Text of 15, to indicate the editing person's name.
6. Discards	Text of 18, to indicate the ingot discards required.
7. Insp._Notice	Text of 18, to indicate the notice time to be given to the inspection authority.
8. Insp.1	Text of 4, to indicate the inspection authority 1.
9. Insp.2	Text of 4, to indicate the inspection authority 2.
10. Insp.3	Text of 4, to indicate the inspection authority 3.
11. Insp.4	Text of 4, to indicate the inspection authority 4.

1-4-1-2 QP_Status

(Already listed in Appendix 1.3)

1-4-1-3 QP_Operations

Description: This relation is designed as the main sub-form for Quality Plan. It records quality plan operational sequences, comments to each operation, document control and inspection instructions.

1. QP_No. Text of 5, the first identifier in the Quality Plan.
2. QP_I Text of 1, the second identifier in the Quality Plan. (QP_No and QP_I are used to group all operations together in one Quality Plan.)
3. Seq. Number 2, for line sequence number for operations. (Fields 1, 2 & 3 are used as the composite primary key in this relation.)
4. Op_No. Number, to indicate the operation number in Quality Plan.
5. Op_Comment Text of 56, to show the operation and comment.
6. Docu._Ref. Text of 18, to show the relevant reference document.
7. Docu._Rec. Text of 18, to show the record document.
8. Inspect1 Text of 4, to indicate the first inspection authority requirement for this operation.
9. Inspect2 Text of 4, to indicate the second inspection authority requirement for this operation.
10. Inspect3 Text of 4, to indicate the third inspection authority requirement for this operation.
11. Inspect4 Text of 4, to indicate the fourth inspection authority requirement for this operation.

1-4-2 Forging Procedures

1-4-2-1 FP (Forging Procedure)

Description: This relation is designed to record forging procedures. It is used as the main form to draw information from other relations. It also records the manufacturing plan issue letter to which the forging procedure is issued in order to link these two plans together.

1. FP_No. Text of 5, the first identifier in this relation. (It has the same value as the MP_No. to which the FEL Order Numbers apply.
2. FP_I Text of 1, the second identifier in this relation.
3. Letter Procedure Letter, the third identifier. (The composite of FP_No., FP_I and Letter is the primary key in this relation.)
4. MP_I The Manuf. Plan Issue when the Quality Plan was issued.
5. Date_Edi Date, the latest editing date.
6. Initial Text of 15, to indicate the editing person's name.

1-4-2-2 FP_Status

Description: This relation is used to record forging status such as identifiers, ST_No., Version and Ingot_No., that can uniquely identify each piece of ingots to be forged.

1. FP_No. Text of 5, the first identifier in the FP_Status.
2. FP_I Text of 1, the second identifier in the FP_Status.
3. Letter Procedure Letter, third identifier in the FP_Status. (The composite of FP_No., FP_I and Letter is used to group all the ingots in one Forging Procedure (FP).)

4. ST_No.	Num.String 6, the first identifier in the Steel Order.
5. Version	Text of 1, the second identifier in the Steel Order.
6. Ingot_No.	Text of 1, showing the ingot number in the steel order. (The composite of ST_No., Version and Ingot_No. is the primary key in Ingot_Status relation.)

1-4-2-3 FP_Comments

Description:	This relation is designed as one of sub-forms for Forging Procedure. It records forging sequences and instructions.

1. FP_No.	Text of 5, the first identifier in the FP_Status.
2. FP_I	Text of 1, the second identifier in the FP_Status.
3. Letter	Procedure Letter, the third identifier in the FP_Status. (Those three fields are used to group all the comment records in one FP.)
4. Seq.	Number, to indicate the line sequence number for the FP comments. (Those four fields above can be used as the composite primary key in this relation.)
5. Comment	Text of 56, for any comment on the procedure.

1-4-3 Heat Treatment Procedures

1-4-3-1 HTP (Heat Treatment Procedure)

Description:	This relation is designed to record heat treatment procedures. It is used as the main form to draw information from other relations. It also records the manufacturing plan issuing to which the forging procedure is issued in order to link these two plans together.

1. HTP_No.	Text of 5, the first identifier in the relation. (Its value is generated from Manuf. Plan for the same target FEL Order Numbers.)
2. HTP_I	Text of 1, the second identifier in the relation.
3. Letter	Procedure Letter, the third identifier in the relation. (The composite of those three fields above is the primary key in this relation.)
4. MP_I	The Manuf. Plan Issue when the HTP was issued.
5. Date_Edi	Date, showing the editing date.
6. Initial	Text of 15, showing the editing person's name.
7. Processes	Text of 14, to indicate the heat treatment processes.

1-4-3-2 HTP_Status

Description:	This relation is used to record heat treatment status such as FEL Order Numbers that are grouped to go through the same heat treatment.

1. HTP_No.	Text of 5, the first identifier in the HTP.
2. HTP_I	Text of 1, the second identifier in the HTP.
3. Letter	Text of 1, procedure letter, the third identifier in the HTP. (The composite of HTP_No., HTP_I and Letter is used to group all target FEL Order Numbers in one HTP.)
4. FEL Order No.	Num.String of 11, showing the target FEL Order No.(s). (It is different from the FEL Order No. in Ingot_Status and is grouped in general for certain numbers according to the steel forging cast or cut.)

1-4-3-3 HTP_Comments

Description: This relation is designed as one of sub-forms for Heat Treatment Procedure. It records all heat treatment cycles and comments.

1. HTP_No. Text of 5, the first identifier in the HTP.
2. HTP_I Text of 1, the second identifier in the HTP.
3. Letter Text of 1, procedure letter, the third identifier in the HTP.
(The composite of those fields above is used to group all comments in one HTP.)
4. Seq. Num. string of 2, Line Sequence number.
(Those four fields above can be used as the composite primary key in this relation.)
5. Stage Number, showing the stage of the heat treatment process.
6. Heat_Rate Text of 6, showing the heating rate in ° C/HR.
7. Temp1 Text of 4, showing the Min. temperature to reach.
8. Temp2 Text of 4, showing the Max. temperature to reach.
9. Hold_1 Number, showing the holding hours at the above temperature range.
10. Medium Choice of cooling medium.
11. Cool_rate Text of 6, showing the cooling rate in ° C/HR.
12. Q_Time Text of 5, showing the quench time to be taken.
13. Cool_To1 Text of 4, showing the Min. temperature to cool.
14. Cool_To2 Text of 4, showing the Max. temperature to cool.
15. Hold_2 Number, showing the holding hours at the above temperature range.
16. Remarks Text of 56, for comment on the heat treatment.

1-4-4 Clean, Packing and Protection Procedures

1-4-4-1 CPP (Clean, Packing and Protection Procedure)

Description: This relation is designed to record Clean, Packing and Protection (CPP) Procedures. It is used as main the form to draw information from other relations. It also records the manufacturing plan issuing to which the CPP procedure is issued in order to link these two plans together.

1. CPP_No. Text of 5, the first identifier in the CPP procedure. It has the same value as the MP_No. that the FEL Order Numbers apply.
2. CPP_I Text of 1, the second identifier in the CPP procedure.
(The composite of CPP_No. and CPP_I is the primary key in this relation.)
3. MP_I The Manuf. Plan Issue when the CPP was issued.
4. Date_Edi Date, showing the editing date.
5. Initial Text of 15, showing the latest editing person's name.

1-4-4-2 CPP_Comments

Description: This relation is designed as a sub-form for CPP Procedure. It records all sequence and instructions for the CPP Procedure.

1. CPP_No. Text of 5, the first identifier in the CPP.
2. CPP_I Text of 1, the second identifier in the CPP.
(Those two fields are used to group all comments in one FP Procedure.)
3. Seq. Line sequence number.
(Those three fields above can be used as the composite primary key in this relation.)
4. Comment Text of 56, for the CPP comment.

1-4-5 Mechanical Testing Procedures

1-4-5-1 MT (Mechanical Testing Procedure)

Description:		This relation is designed to record Mechanical Testing (MT) Procedures. It isused as the main form to draw information from MT_Comments. It also records the manufacturing plan issuing to which the MT Procedure is issued in order to link these two plans together.
1.	MT_No.	Text of 5, the first identifier in the MT procedure. Its value is same as the MP_No. that the FEL Order Numbers apply.
2.	MT_I	Text of 1, the second identifier in the MT procedure. (The composite of MT_No. and MT_I is the primary key in this relation.)
3.	MP_I	The Manuf. Plan Issue when the MT was issued.
4.	Date_Edi	Date, showing the editing date.
5.	Initial	Text of 15, showing the latest editing person's name.

1-4-5-2 MT_Comments

Description: This relation is designed as sub-form for MT Procedure. It records all sequence and instructions for the MT Procedure.

1. MT_No. — Text of 5, the first identifier in the MT procedure.
2. MT_I — Text of 1, the second identifier in the MT procedure. (These two fields are used to group all comments in one FP Procedure.)
3. Seq. — Integer number, to indicate line sequence number. (These three fields above may be used as the composite primary key in this relation.)
4. Comment — Text of 56, to show the MT comment.

1-4-6 Special Procedures

1-4-6-1 SP (Special Procedure)

Description: This relation is designed to record Special Procedures. It is used as main form to draw information from other relation such as SP_Comments. It also records the manufacturing plan issue to which the Special Procedure is issued in order to link these two plans together.

1. SP_No. — Text of 5, the first identifier in the SP procedure. Its value is taken from MP_No. to which the FEL Order Numbers apply.
2. SP_I — Text of 1, the second identifier in the SP procedure. (The composite of SP_No. and SP_I is the primary key in this relation.)
3. MP_I — The Manuf. Plan Issue number when the SP is issued.
4. Date_Edi — Date, showing the latest editing date.
5. Initial — Text of 15, showing the latest editing person's name.

1-4-6-2 SP_Comments

Description: This relation is designed as sub-form for SP Procedure. It records all sequence and instructions for the special procedure.

1. SP_No. Text of 5, the first identifier in the SP procedure.
2. SP_I Text of 1, the second identifier in the SP procedure.
   (These two fields are used to group all comments in one SP Procedure.)
3. Seq. Integer number, to indicate line sequence number.
   (These three fields above may be used as the composite primary key in this relation.)
4. Comment Text of 56, to show the SP comment.

1-4-7 Control Document Status Lists

1-4-7-1 CDSL (Control Document Status List)

Description: This relation is designed to record Control Document Status Lists. It is used as the main form to draw information from other relation such as CDSL_Comments. It also records the manufacturing plan issuing to which the CDSL Procedure is issued in order to link these two plans together.

1. CDSL_No. Text of 5, the first identifier in the CDSL procedure. Its value is taken from the MP_No. to which the FEL Order Numbers apply.
2. CDSL_I Text of 1, the second identifier in the CDSL.
   (The composite of CDSL_No. and CDSL_I is the primary key in this relation.)
3. MP_I The Manuf. Plan Issue when the CDSL was issued.
4. Date_Edi Date, showing the latest editing date.
5. Initial Text of 15, showing the latest editing person's name.

1-4-7-2 CDSL_Comments

Description: This relation is designed as a sub-form for CDSL Procedure. It records all related documents, document titles and revision status.

1. CDSL_No. Text of 5, the first identifier in the CDSL.
2. CDSL_I Text of 1, the second identifier in the CDSL.
   (These two fields are used to group all the records in one CDSL Procedure.)
3. Seq. Integer number, to indicate line sequence number.
   (These three fields above may be used as the composite primary key in this relation.)
4. Document Text of 20, showing the name of the documents listed.
5. Title Text of 40, showing the document title.
6. Revision Text of 12, showing the issue of the revision.

1-5 Fields in the Relations in the Steel Order Plan Planning Area

1-5-1 Steel Orders

1-5-1-1 Steelord

Description: This relation records the main part of Steel Order information. It indicates the approval of the Steel Order, the current related Manufacturing Plan, and the designed melting range.

1. ST_No. Num.String 6, the first identifier in this relation.
2. Version Text of 1, the second identifier in this relation.

	(The composite of ST_No. and Version is the primary key in this relation.)
3. Approve	Choice of Yes/No, to indicate whether the steel Order has been approved by the metallurgist or not.
4. MP_No.	Text of 5, showing the related Manuf. Plan Number.
5. Issue	Text of 1, showing the related Manuf. Plan Issue. (MP_No. and Issue are used to obtain information from the related Manuf. Plan for printing or viewing purposes.)
6. To_Producer	Text of 20, to indicate the producer of the steel.
7. Date	Date, to indicate the latest editing date.
8. Initial	Text 15, showing the latest editing person's name.
9. Melting_No.	Text of 5, to indicate the related Melting Range. (Melting_No. is the primary key in the Melt._Range relation.)
10. Hot_To_Forge	Choice of Yes/No, to indicate whether the ingot needs to be delivered in hot.
11. Chemistry	Choice of Yes/No, to indicate whether the cast analysis needs to be sent to FEL by Fax..
12. Melt.Racord	Choice of Yes/No, to indicate whether the melting record needs to be recorded.
13. Inc.Analysis	Choice of Yes/No, to indicate whether the certificate of confirmation is required.
14. Vacuum Degas	Choice of Yes/No, to indicate whether vacuum degassing is required.
15. Deoxidisation	Choice to indicate the deoxidisation methods to be used.
16. Teeming	Choice to indicate teeming process to be used.
17. Ultrasonic	Choice of ultrasonic parameters.
18. FEL Note1	Text of 58, for any special comments on the order.
19. FEL Note2	Text of 58, for any special comments on the order.
20. FEL Note3	Text of 58, for any special comments on the order.
21. Print?	Choice of Yes/No, to indicate whether or not the customer chemical range should be printed on the Steel Order report.

1-5-1-2 SO_Status

(Already listed in Appendix 1.4)

1-5-1-3 FEL_Melt.Range

Description:	This relation is designed to record Melting Ranges, each of which having up to 22 elements with Max. and Min. values. It also records the steel quality and designer's comments about the melting range.

1. Melting_No.	Num.String of 5, the primary key in this relation.
2. Name	Text of 15, to indicate who has designed the range.
3. Date	Date, to indicate the latest editing date.
4. Quality	Text of 15, indicating the steel quality for this range.
5. CMin	Num.String, minimum content of Carbon.
6. CMax	Num.String, maximum content of Carbon.
7. SiMin	Num.String, minimum content of Silicon.
8. SiMax	Num.String, maximum content of Silicon.
9. MnMin	Num.String, minimum content of Manganese.
10. MnMax	Num.String, maximum content of Manganese.
11. PMin	Num.String, minimum content of Phosphorus.
12. PMax	Num.String, maximum content of Phosphorous.
13. SMin	Num.String, minimum content of Sulphur.
14. SMax	Num.String, maximum content of Sulphur.
15. CriMin	Num.String, minimum content of Chromium.
16. CrMax	Num.String, maximum content of Chromium.
17. Other elements	(There are total 22 elements with Min. and Max. value fields, similar to Carbon or Silicon. 18 elements are named, four are available for definition by the user.)
18. Imprecise	Text of 60, to indicate imprecise comments.

1-5-2 Ingot Requirements

1-5-2-1 Ingot_Requirements

Description: This relation is designed as the main form to draw information from other relations, Ingot_Status and Ingot. These relations record information related to ingot requirements in the Steel Order. The main form also shows whether or not the ingot requirement has been approved by the designer.

1. ST_No. — Num.String 6, the first identifier in the Steel Order.
2. Version — Text of 1, the second identifier in the Steel Order.
   (The composite of ST_No. and Version is the primary key in both Steelord and Ingot_Requirements relations and used to link those two together.)
3. Approve — Choice of Yes/No, to indicate whether the ingot requirement has been approved by the Forge Planner or not. If not, the Steel Order can not be issued.

1-5-2-2 Ingot_Status

Description: This relation is designed as one of sub-forms for Ingot_Requirement. It shows how FEL Order Numbers are grouped in each type of ingots.

1. ST_No. — Num.String 6, the first identifier in the Steel Order.
2. Version — Text 1, the second identifier in the Steel Order.
   (The composite of ST_No. and Version is used to group all FEL Order Numbers together in one Steel Order.)
3. Ingot_No. — Text of 1, to indicate the ingot sequence number in the steel order.
4. FEL Order No. — Num.String, it is different from the FEL Order No. in Forging Status relation, showing a group of FEL Order Numbers in numeric string according to the ingot size.

1-5-2-3 Ingot

Description: This relation is designed as one of sub-forms for Ingot_Requirement. It gives the design parameters fro each ingot specified in the Ingot_Status file.

1. ST_No. — Num.String 6, the first identifier in the Steel Order.
2. Version — Text of 1, the second identifier in the Steel Order.
   (The composite of ST_No. and Version is used to group all ingots together in one Steel Order.)
3. Ingot_No. — Text of 1, to indicate the ingot number in the steel order.
   (The composite of those three attributes may be used as the primary key in this relation.)
4. Date — Date, to indicate the latest editing date.
5. Initial — Text of 15, name of the person who designs the ingot.
6. Ingot type — Choice of ingot types.
7. Length/Weight — Choice of length or chill weight to be specified.
8. L/W_Value — Text of 4, to show a Value for length or chill weight.
9. Weight — Number, to indicate the ingot total weight in tones.
10. Delivery — Choice of delivery places, such as forge or machine shop.

1-6 Fields in the Relations in the Test Results Function Area

1-6-1 Steelmaking Programme from FSL

1-6-1-1 FSL_Proc.

Description:	This relation is designed as the main relation to store chemical ranges to be achievedby FSL and reported in Steelmaking Programme. It records all 22-elementchemical ranges in the same structure as the Melting Range relation in order for procedures to compare these two ranges to identify any difference.
1. Item_No.	Num.String 5, the first identifier in the relation.
2. Item_Issue	Text of 2, the second identifier in the relation. (Item_No. and Item_Issue are used as the composite primary key in this relation.
3. Quality	Text of 15, showing the steel quality designed.
4. RMC	Text of 7, showing the RMC code to be used.
5. CMin	Num.String, minimum content of Carbon.
6. CMax	Num.String, maximum content of Carbon.
7. SiMin	Num.String, minimum content of Silicon.
8. SiMax	Num.String, maximum content of Silicon.
9. MnMin	Num.String, minimum content of Manganese.
10. MnMax	Num.String, maximum content of Manganese.
11. PMin	Num.String, minimum content of Phosphorus.
12. PMax	Num.String, maximum content of Phosphorous.
13. SMin	Num.String, minimum content of Sulphur.
14. SMax	Num.String, maximum content of Sulphur.
15. CriMin	Num.String, minimum content of Chromium.
16. CrMax	Num.String, maximum content of Chromium.
17. Other Elements	(There are totally 22 elements with Min. and Max. value fields, similar to Carbon or Silicon. 18 elements are pre-defined, four are available for definition by users.)

1-6-1-2 FSL_Ingot

Definition:	This relation records ingot design parameters specified in the Steelmaking Programmes from FSL. It is used as a sub-relation to show all ingots for the main form, FSL_Proc.
1. Item_No.	Num.String 5, the first identifier in the FSL_Proc..
2. Item_Issue	Text of 2, the second identifier in the FSL_Proc.. (Item_No. and Item_Issue are used to group all ingots designed by FSL together in one steelmaking programme.)
3. Order_No.	Num.String, FEL Order Number string.
4. No.	Text of 1, to indicate the ingot number in sequence. (The composite of those three attributes can be used as the primary key in this relation.)
5. Ingots	Choice of ingot types.
6. Weight	Number, to indicate the ingot total weight in tones.
7. Dest'n	Choice of delivery places.
8. Delivery	Text of 15, showing delivery conditions.
9. Time_Due	Text of 12, indicating the date that the ingot is to be cast.

1-6-1-3 FSL_Programme

Description: This relation is designed as a sub-relation to the main form of FSL_Proc, to store steelmaking processes to be taken by FSL and reported in the Steelmaking Programme.

1. Item_No. — Num.String 5, the first identifier in the relation.
2. Item_Issue — Text of 2, the second identifier in the relation. (Item_No. and Item_Issue are used as the composite primary key in the relation, which is same as the primary key in FSL_Proc. relation.)
3. Scrap — Choice of scrap preparation conditions.
4. Lad._No. — Choice of ladle numbers.
5. Liquid — Number, showing liquid weight in tones.
6. Spare — Number, showing spare liquid metal in tones.
7. Process1 — Choice 1 for selection of processes.
8. Process2 — Choice 2 for selection of processes.
9. Slag — Choice of slag practices
10. Slag_Wt. — Number, slag weight in Kilos.
11. Deoxidisation — Choice of deoxidisation.
12. Temperatures — Text of 4, showing certain temperatures including liquidus and aiming temperatures.
13. Hydrogen — Choice of Yes/No, showing whether a hydrogen sample is to be taken.
14. Mould — Text of 12, showing mould conditions.
15. Tiles — Choice of tile types and lengths.
16. Nozzle — Choice of nozzle sizes.
17. Head — Choice of conditions that ingot head is to be filled to.
18. Powder — Choice of heading powders
19. Stand_Time — Choice of standing time in hours.
20. Others — Text of long comments.

1-6-2 Cast Analysis from FSL

1-6-2-1 Analysis_Index

Description: This relation is designed as the main form to group a series of cast analysis taken from different positions for the same ingot.

1. Cast_No. — Text of 5, the first identifier in this relation.
2. Cast_Letter — Text of 5, the second identifier in this relation. (Cast_No. and Cast_Letter are used as the composite primary key in the relation.)
3. Date — Date, that the ingot was cast.

1-6-2-2 Analysis

Description: This relation is used to record cast analysis for up to 22 element. It indicates position from which the analysis is taken.

1. Cast_No. — Text of 5, the first identifier in Analysis_Index.
2. Cast_Letter — Text of 5, the second identifier in Analysis_Index. (Cast_No. and Cast_Letter are used to group all analyses taken from different positions together in one Cast Analysis report.)
3. Cat's — Choice of positions.
4. RWS — Choice of Yes/No, shows whether or not the cast analysis is within the FEL requirements checked by system checking procedures.

5. CMin — Num.String, minimum content of Carbon.
6. CMax — Num.String, maximum content of Carbon.
7. SiMin — Num.String, minimum content of Silicon.
8. SiMax — Num.String, maximum content of Silicon.
9. MnMin — Num.String, minimum content of Manganese.
10. MnMax — Num.String, maximum content of Manganese.
11. PMin — Num.String, minimum content of Phosphorus.
12. PMax — Num.String, maximum content of Phosphorous.
13. SMin — Num.String, minimum content of Sulphur.
14. SMax — Num.String, maximum content of Sulphur.
15. CriMin — Num.String, minimum content of Chromium.
16. CrMax — Num.String, maximum content of Chromium.
17. Other Elements — (There are totally 22 other elements with Min. and Max. value fields, similar to Carbon or Silicon. 18 elements are named, four are available for definition by the user.)

1-6-3 NDT Reports

1-6-3-1 U/S_Report

1-6-3-1-1 U/S_Report

Description: This relation is designed as the main form to record Ultrasonic (U/s) test reports.

1. NDT_No. — Text of 5, NDT Number.
2. Op_No. — Integer number of 2, to indicate operation number in manufacturing plan.
3. Batch — Integer number of 2, to indicate batch number of production of forgings. (Both NDT_No., Op_No. and Batch are the composite primary key in the U/S_Report relation.)

4. FEL_Order_No. — String format of FEL Order Numbers
5. Quantity — Integer number to indicate numbers of forgings being tested.
6. Identity — Text of 8, to indicate cast Identifications
7. ID_No. — Text of 5, to indicate Identification record at the time the test conducted.
8. Ver. — Text of 1, to indicate identification record issue number at the time the test conducted.
9. Operator — Text of 20, to indicate operator name.
10. Qualification — Text of 18, to indicate operator's qualification at the time the test conducted.
11. Date — Date showing when the test is conducted.
12. Stage — Choice of stages to indicate whether the test is at intermediate or final stage.
13. NDT_Spec.1 — Text of 30, to indicate NDT specifications used.
14. NDT_Spec.2 — Text of 30, to indicate NDT specifications used.
15. NDT_Spec.3 — Text of 30, to indicate NDT specifications used.
16. Surface — Choice of surface finish quality, 125 CLA or 250 CLA.
17. Detector_No. — Foreign determinant key from relation of NDT_Instrument.
18. Release — Yes or no, to show whether the report has been released to the customer.
19. Couplant — Choices of scanning couplants, such as SAE 30 oil, Cellulose paste and ultragel 2.
20. Note1 — Text of 69, to record examination result.
21. Note2 — Text of 69, to record examination result.
22. Note3 — Text of 69, to record examination result.
23. Results — Choice of results, such as "Indications observed see attached page(s) No reportable indications observed, etc."
24. Verdict — Text of 42, to record verdict from operator.
25. Evaluator — Text of 15, to record evaluator's verdict.

26. Sketch — Yes or no, to indicate whether a sketch is attached for details of indications found.
27. Date2 — Date for the final verdict.

1-6-3-1-2 U/S_Scan

Description: This relation is designed as a sub-form to record U/S scanning results.

1. NDT_No. — Text of 5, NDT Number.
2. Op_No. — Number of 2 to indicate operation number in manufacturing plan.
3. Batch — Number of 2 to indicate batch number of production of forgings. (Both NDT_No., Op_No. and Batch are the composite primary key in the U/S_Report relation, and to group all scanning result for the same identifiers.)
4. Scan_No. — Number field to indicate scanning sequence number.
5. Direction — Choice of scan directions, such as Longitudinal, Radial and Transverse.
6. Probe_No. — Text of 5, to indicate probe number used for the scan, foreign key in relation, Probe.
7. Text1 — Choice of scan processes, such as "D. A. C. Curve from" or "1st backwall echo to" etc.
8. Value1 — Number of 4, to indicate values for the above scan process in Text1.
9. Text2 — Choice of scan processes, such as "mm Dia. F.B.H's" or "% F. S. H." etc.
10. Value2 — Integer number of 2, to indicate values for the above scan process in Text 2.
11. Plus — Text of 1, to indicate "Plus" or "Miner" sign to the value in Value3 field below.
12. Value3 — Number of 2, to indicate the values for the scan process specified in Text3 field below.
13. Text3 — Choice of scan processes, such as "mm grass level at the maximum test distance" or "mm Dia. F.B.H. ref. sensitivity at the axis as per the AVG" etc.

1-6-3-1-3 NDT_Instrument

Description: This relation is used to record instrument type. Each type is identified by the Instrument Number.

1. Instrument_No. — Text of 8, to identify unique detector.
2. Istrument_Type — Text of 20, to indicate instrument type.

1-6-3-1-4 Probe

Description: This relation is used to record probe parameters.

1. Probe_No. — Text of 5, to indicate probe number, primary identifier in the relation.
2. Wave — Text of 10, to indicate probe waves.
3. Angle_deg. — 2 digit Integer Number, to indicate the angle degree of probe.
4. Probe_Type — Text of 15, to indicate probe type.
5. FreqMHz — 4 digit fixed point number, to indicate probe frequency.
6. S/Twin — Choice of probe type, such as Single or Twin.
7. Crystal_Size — 4 digit fixed point number to indicate crystal sizes.

1-6-3-2 MPI Report

1-6-3-2-1 MPI_Report

Description: This relation is designed as the main form to record Magnetic Particle Inspection (MPI) reports.

1. NDT_No.	Text of 5, NDT Number.
2. Op_No.	Number of 2 to indicate operation number in manufacturing plan.
3. Batch	Number of 2 to indicate batch number of production of forgings. (Both NDT_No., Op_No. and Batch are the composite primary key in the MPI_Report relation.)
4. FEL_Order_No.	String format of FEL Order Numbers
5. Quantity	Integer number to indicate numbers of forgings being tested.
6. Identity	Text of 8, to indicate cast Identifications
7. ID_No.	Text of 5, to indicate Identification record at the time the test conducted.
8. Ver.	Text of 1, to indicate identification record issue number at the time the test conducted.
9. Operator	Text of 20, to indicate operator name.
10 Qualification	Text of 18, to indicate operator's qualification at the time the test conducted.
11. Date	Date showing when the test is conducted.
12. Stage	Choice of stages to indicate whether the test is at intermediate or final stage.
13. NDT_Spec.1	Text of 30, to indicate NDT specifications used.
14. NDT_Spec.2	Text of 30, to indicate NDT specifications used.
15. NDT_Spec.3	Text of 30, to indicate NDT specifications used.
16. Surface	Choice of surface finish quality, 125 CLA or 250 CLA.
17. Instrument_No.	Foreign determinant key in relation of NDT_Instrument.
18. Method	Choices, to show method of magnetisation, such as Residual or Continuous.
19. Technique	Choice of techniques of magnetisation, such as Coils, Prods, headstock and central conductor etc.
20. Direction	Choice of magnetisation directions, such as Longitudinal, Circular, or 2 mutually perpendicular directions.
21. Prod_Tip	Choice of prod tips, such Aluminium, Copper, lead, mild steel or mild steel pole pieces.
22. Turns	Text of 1, to indicate number of turns .
23. Between	Choice of distances, such as Prods or York Legs.
24. Space	Text of 10, to indicate test conditions.
25. Unit1	Choice of Amp, inches or mm.
26. MPE	Choice of Magnetising Amperes.
27. Capable	Choice of "Capable of lifting a 4.5 Kg weight at a pole spacing of" or "Capable of lifting a 10 Kg weight at a pole spacing of".
28. Value1	Integer number, to indicate the value for the above conditions.
29. Unit2	Choice of Amp, inches or mm, the unit for the above value.
30. Type	Choice of FWDC, HWDC or AC, to indicate current type.
31. Particle	Choice of "WET paraffin based (black)", Wet paraffin based (fluorescent)", or "Water based", to indicate magnetising particle type.
32. Long:Detail1	Text of 78, to record details of examination.
33. Long:Detail2	Text of 78, to record details of examination.
34. Long:Detail3	Text of 78, to record details of examination.
35. Results	Choice of results, such as "Indications observed see attached page(s) No reportable indications observed, etc."
36. Verdict	Text of 42, to record verdict from operator.
37. Evaluator	Text of 15, to record evaluator's verdict.
38. Sketch	Yes or no, to indicate whether a sketch is attached for details of indications found.
39. Date2	Date for the final verdict.

1-6-3-2-2 NDT_Instrument

As listed in 1-6-3-1-3

1-6-3-3 Dye_Pen_Report

Description:	This relation is designed as the main form to record Dye Penetrate (Dye_Pen) test reports.
1. NDT_No.	Text of 5, NDT Number.
2. Op_No.	Number of 2 to indicate operation number in manufacturing plan.
3. Batch	Number of 2 to indicate batch number of production of forgings. (Both NDT_No., Op_No. and Batch are the composite primary key in the Dye_Pen_Report relation.)
4. FEL_Order_No.	String format of FEL Order Numbers
5. Quantity	Integer number to indicate numbers of forgings being tested.
6. Identity	Text of 8, to indicate cast Identifications
7. ID_No.	Text of 5, to indicate Identification record at the time the test conducted.
8. Ver.	Text of 1, to indicate identification record issue number at the time the test conducted.
9. Operator	Text of 20, to indicate operator name.
10. Qualification	Text of 18, to indicate operator's qualification at the time the test conducted.
11. Date	Date showing when the test is conducted.
12. Spec.1	Text of 30, to indicate NDT specifications used.
13. NDT_Spec.2	Text of 30, to indicate NDT specifications used.
14. NDT_Spec.3	Text of 30, to indicate NDT specifications used.
15. Stage	Choice of stages to indicate whether the test is at intermediate or final stage.
16. Surface	Choice of surface finish quality, 125 CLA or 250 CLA.
17. Penetrant	Text of 20, to indicate the Penetrant used.
18. Type	Choice of Water Washable, Solvent Removable, Post Emulsified or Fluorescent, to indicate the Penetrant type.
19. AMB_Temp	Integer Number of 3, to indicate the test temperature.
20. Dwell_Time	Time for dwelling.
21. View_Time	Time for viewing.
22. Long:Detail1	Text of 78, to record details of examination.
23. Long:Detail2	Text of 78, to record details of examination.
24. Long:Detail3	Text of 78, to record details of examination.
25. Results	Choice of results, such as "Indications observed see attached page(s) No reportable indications observed, etc."
26. Verdict	Text of 42, to record verdict from operator.
27. Evaluator	Text of 15, to record evaluator's verdict.
28. Sketch	Yes or no, to indicate whether a sketch is attached for details of indications found.
29. Date2	Date for the final verdict.

1-6-3-4 Visual_Report

Description:	This relation is designed as the main form to record Visual test reports.
1. NDT_No.	Text of 5, NDT Number.
2. Op_No.	Number of 2 to indicate operation number in manufacturing plan.
3. Batch	Number of 2 to indicate batch number of production of forgings. (Both NDT_No., Op_No. and Batch are the composite primary key in the Visual_Report relation.)
4. FEL_Order_No.	String format of FEL Order Numbers

5. Quantity — Integer number to indicate numbers of forgings being tested.
6. Identity — Text of 8, to indicate cast Identifications
7. ID_No. — Text of 5, to indicate Identification record at the time the test conducted.
8. Ver. — Text of 1, to indicate identification record issue number at the time the test conducted.
9. Operator — Text of 20, to indicate operator name.
10. Qualification — Text of 18, to indicate operator's qualification at the time the test conducted.
11. Date — Date showing when the test is conducted.
12. NDT_Spec.1 — Text of 30, to indicate NDT specifications used.
13. NDT_Spec.2 — Text of 30, to indicate NDT specifications used.
14. NDT_Spec.3 — Text of 30, to indicate NDT specifications used.
15. Stage — Choice of stages to indicate whether the test is at intermediate or final stage.
16. Surface — Choice of surface finish quality, 125 CLA or 250 CLA.
17. Long:Detail1 — Text of 78, to record details of examination.
18. Long:Detail2 — Text of 78, to record details of examination.
19. Long:Detail3 — Text of 78, to record details of examination.
20. Results — Choice of results, such as "Indications observed see attached page(s) No reportable indications observed, etc."
21. Verdict — Text of 42, to record verdict from operator.
22. Evaluator — Text of 15, to record evaluator's verdict.
23. Sketch — Yes or no, to indicate whether a sketch is attached for details of indications found.
24. Date2 — Date for the final verdict.

1-6-3-5 S_Print_Report

Description: This relation is designed as the main form to record Sulphur Print test reports.

1. NDT_No. — Text of 5, NDT Number.
2. Op_No. — Number of 2 to indicate operation number in manufacturing plan.
3. Batch — Number of 2 to indicate batch number of production of forgings. (Both NDT_No., Op_No. and Batch are the composite primary key in the S_Print_Report relation.)
4. FEL_Order_No. — String format of FEL Order Numbers
5. Quantity — Integer number to indicate numbers of forgings being tested.
6. Identity — Text of 8, to indicate cast Identifications
7. ID_No. — Text of 5, to indicate Identification record at the time the test conducted.
8. Ver. — Text of 1, to indicate identification record issue number at the time the test conducted.
9. Operator — Text of 20, to indicate operator name.
10. Qualification — Text of 18, to indicate operator's qualification at the time the test conducted.
11. Date — Date showing when the test is conducted.
12. Stage — Choice of stages to indicate whether the test is at intermediate or final stage.
13. Long:Detail1 — Text of 78, to record details of examination.
14. Long:Detail2 — Text of 78, to record details of examination.
15. Long:Detail3 — Text of 78, to record details of examination.
16. Results — Choice of results, such as "Indications observed see attached page(s) No reportable indications observed, etc."
17. Verdict — Text of 42, to record verdict from operator.
18. Evaluator — Text of 15, to record evaluator's verdict.
19. Sketch — Yes or no, to indicate whether a sketch is attached for details of indications found.
20. Date2 — Date for the final verdict.

1-6-4 Mechanical Property Test Results

1-6-4-1 Certificate

Description:	This relation is designed as the main form to record the main part of Mechanical Property Test Results. It also indicates satisfaction of the report and verdict given by the shop floor.
1. Certi_No.	Text of 5, test certificate number.
2. Batch	Number of 2, to indicate batch number of production of forgings. (Both Certi_No. and Batch are the composite primary key in the Certificate relation and used to group the mechanical properties for the same Certi_No. and Batch.)
3. Test_Date	Date, shows the date that the test conducted.
4. FEL Order No.	Pre-defined string format of FEL Order Numbers.
5. Cast_No.	Text of 8, to indicate cast identifications.
6. T_Unit	Choice of tensile units, such as MPa, N/sq.mm, MN/Sq.m, tonf/Sq.in, lbf/Sq.in, Kgf/Sq.mm or h bars.
7. I_Unit	Choice of Impact units, such as J, N m, ft lbf, Kgf m, or J/Sq.cm.
8. Stress	Choice of stress types, such as Rp0.1, Rp0.2, Rp0.5, Rp, Rp1.0, or Rp0.01.
9. T_Name	Choice of temperature units, such as C or F.
10. Long:Detail1	Text of 78, to record details of other tests.
11. Long:Detail2	Text of 78, to record details of other tests.
12. Long:Detail3	Text of 78, to record details of other tests.
13. Results	Text of 31 to indicate satisfaction of the results.
14. Verdict	Text of 42, to record verdict from QA.
15. Date2	Date for the final verdict.

1-6-4-2 Mech_Tests

Description:	This relation is used as a sub-form to record mechanical test results taken on different directions or orientations.
1. Certi_No.	Text of 5, test certificate number.
2. Batch	Number of 2, to indicate batch number of production of forgings. (Both FES and Group are the composite primary key in the Specification relation and used to group the mechanical properties for the same FES and Group.)
3. Dir.	Choice of directions, e.g. axial, radial etc. to show the direction in which the test should be taken.
4. Location	Text of 6, to show the location of the test sample.
5. Section	Text of 6, to show the section from which the test sample will be taken.
6. Ten_Temp.	Text of 4, to show tensile temperature.
7. Ten_Unit	Choice of units in which tensile test to be recorded.
8. Ten_Value 1	Text of 5, to show the minimum tensile test value.
9. Ten_Value 2	Text of 5, to show the maximum tensile test value.
10. RM1	Text of 5, to show the minimum RM value.
11. RM2	Text of 5, to show the maximum RM value.
12. A.	Text of 3, showing minimum percentage elongation after fracture.
13. Z.	Text of 3, showing minimum reduction in area.
14. Imp_Temp.	Text of 4, to show the impact test temperature.
15. Imp_Unit	Choice of test units in which impact test to be recorded.
16. Imp_Value 1	Text of 5, to show the impact test value.
17. Imp_Value 2	Text of 5, to show the reference impact test value.
18. Bend_Angle	Text of 3, to indicate the bend test angles.
19. Bend_Rad.	Text of 3, to indicate the bend test radius.

20. Hardness 1	Text of 4, to show the minimum hardness test value.
21. Hardness 2	Text of 4, to show the maximum hardness test value.
22. Shear	Text of 3, to show the shear percentage value.
23. FATT	Text of 4, to show the FATT value.
24. Remarks	Text of 28, for comment on each mechanical property result.

1-7 Fields in the Relations in the QA Functions Area

1-7-1 Document Distribution Matrix

1-7-1-1 Docu_Ditribution

Description: This relation is designed as the main form to record Document Distribution Lists. It is linked with other sub-forms to record complete status lists.

1. Document — Choice of documents, such as Quality plan, Manufacturing Plan, Steel Order etc.
2. Document_No. — Text of 15, to indicate document number.
   (Both document and Document_No. are the composite primary key in the Docu_Distribution relation and used to group the distribution matrix for the same identifiers.)
3. Cleared — Yes or no field, to indicate whether all documents have been withdrew.
4. Issue1 — Text of 1, to indicate the relevant issue for the document distributed.
5. Issue2 — Text of 1, to indicate the relevant issue for the document distributed.
6. Issue3 — Text of 1, to indicate the relevant issue for the document distributed.
7. Issue4 — Text of 1, to indicate the relevant issue for the document distributed.
8. Issue5 — Text of 1, to indicate the relevant issue for the document distributed.
9. Issue6 — Text of 1, to indicate the relevant issue for the document distributed.
10. Issue7 — Text of 1, to indicate the relevant issue for the document distributed.
    (These 7 issues are used for circulated document and withdraw matrix.)

1-7-1-2 Issue_Status

Description: This relation is used as a sub-form to record Issue Status for each report of the Document Distribution List.

1. Document — Choice of documents, such as Quality plan, Manufacturing Plan, Steel Order etc.
2. Document_No. — Text of 15, to indicate document number.
   (Both document and Document_No. are the composite primary key in the Docu_Distribution relation and used to group the issue status for the same identifiers.)
3. Issue — Text of 1, to indicate issue letter for the document.
4. Issue_Date — Date to record when the document is issued and distributed.
5. Note — Text of 20, to record necessary information for that issue.

1-7-1-3 Distr_Matrix

Description: This relation is used as a sub-form to record Distribution Matrix for each report of Document Distribution Lists.

1. Document — Choice of documents, such as Quality plan, Manufacturing Plan, Steel Order etc.
2. Document_No. — Text of 15, to indicate document number.
   (Both document and Document_No. are the composite primary key in the Docu_Distribution relation and used to group the distribution matrix for the same identifiers.)
3. No. — Integer number to indicate the sequence of the departments.

4. Distribute_To — Choice of departments, such as Sales, Metallurgists, Forge Planning etc.
5. Name — Text of 5, to record initials of the authority person in the department.
6. Copy — Integer number of 2, to record the numbers of document distributed to that department.
7. Issue1 — Yes or no field, to indicate whether or not relevant issue of document has been withdrawn.
8. Issue2 — Yes or no field, to indicate whether or not relevant issue of document has been withdrawn.
9. Issue3 — Yes or no field, to indicate whether or not relevant issue of document has been withdrawn.
10. Issue4 — Yes or no field, to indicate whether or not relevant issue of document has been withdrawn.
11. Issue5 — Yes or no field, to indicate whether or not relevant issue of document has been withdrawn.
12. Issue6 — Yes or no field, to indicate whether or not relevant issue of document has been withdrawn.
13. Issue7 — Yes or no field, to indicate whether or not relevant issue of document has been withdrawn.
(These 7 fields are used to record distribution matrix for issues specified in Docu_Distribution)

1-7-2 Document sent

1-7-2-1 Docu_Sent

Description: This relation is designed as the main form to record Document Issue Status to customers. It is linked with sub-form, Sent_Status, to list complete documents, issues and response status for quality control.

1. Document — Choice of documents, such as Quality plan, Manufacturing Plan, Steel Order etc.
2. Document_No. — Text of 15, to indicate document number.
(Both document and Document_No. are the composite primary key in the Docu_Sent relation.)
3. Customer_No. — Text of 8, to identify the customer in the Customer relation.
4. Cleared — Yes or no field, to indicate whether or not all documents are subject to the state of re-issue or not-returned.

1-7-2-2 Sent_Status

Description: This relation is used as a sub-form to record issue status for each order specified in the main form, Docu_Sent.

1. Document — Choice of documents, such as Quality plan, Manufacturing Plan, Steel Order etc.
2. Document_No. — Text of 15, to indicate document number.
(Both document and Document_No. are use to group all documents sent.)
3. No. — Integer of 2, sequence number.
4. Issue — Text of 1, document issue letter.
5. Date — Date of sending.
6. Status — Choice of "Awaiting re-issue", "Approved" or "Not-Returned".
7. Note — Text of 40, to record some information.

APPENDIX 2

LISTING OF RELATIONSHIP DEFINITIONS

Relationships are definitions between relations or forms to connect them together within a database. It is used to relate relations to share data, which can allow one form to access the data in another form, produce reports that involve more than one form and display more than one form on the screen at a time. In the following relationship definition, two relations are joined by "AND", and the definition is created by matching fields in both relations.

Steelord AND FEL Range by
    Melting No. = Melting No.

Customer AND Forging_Definition by
    Customer# = Costomer#

FSLproc AND FSLbiginput by
    ITEM = ISSUE

Melting Range AND Status by
    F_No. = FEL Order No.

Modifymanuf AND Status by
    F_No. = FEL Order No.

FSLproc AND FSLingot by
    ITEM = ISSUE

Melting Range AND Steelord by
    FES = Approve

Modifymanuf AND Forging_Definition by
    ID_No. = Ver.

Manuf. Plqan AND RMC-Scrap by
    RMC = RMC Code

Manuf. Plan AND Op._File by
    MP No. = MP No.
    Issue = Issue

U/S_Report AND U/S_Scan by
    NDT_No. = NDT_No.
    Op_No.= Op_No.

Probe AND U/S_Scan by
    Probe_No. = Probe_No.

U/S_Report AND Forging_Definition by
    ID_No. = ID_No.
    Ver. = Ver.

IDN_Meeting AND IDN_Minutes by
    Ref_No. = Ref_No.

IDN_Minutes AND IDN_Report by
    No. = No.

IDN_Report AND Concession by
    No. = No.

U/S_Report AND NDT Report by
    NDT_No. = NDT_No.
    Op_No. = Op_No.
    Batch = Batch

NDT Report AND Status by
    F_No. = FEL Order No.

MPI_Report AND Forging_Definition by
    ID_No. = ID_No.
    Ver. = Ver.

MPI_Report AND NDT Report by
    NDT_No. = NDT_No.
    Op_No. = Op_No.
    Batch = Batch

Dye_Report AND Forging_Definition by
    ID_No. = ID_No.
    Ver. = Ver.

Dye_Report AND NDT Report by
    NDT_No. = NDT_No.
    Op_No. = Op_No.
    Batch = Batch

Visual_Report AND Forging_Definition by
    ID_No. = ID_No.
    Ver. = Ver.

Visual_Report AND NDT Report by
    NDT_No. = NDT_No.
    Op_No. = Op_No.
    Batch = Batch

Sulphur_Report AND Forging_Definition
ID_No. = ID_No.
Ver. = Ver.

Sulphur_Report AND NDT Report by
NDT_No. = NDT_No.
Op_No. = Op_No.
Batch = Batch

U/S_Report AND H_U/S_Scan by
NDT_No. = NDT_No.
Op_No. = Op_No.
Batch = Batch

Probe AND H_U/S_Scan by
Probe_No. = Probe_No.

U/S_Report AND NDT_Instrument by
Detector# = Number

MOI_Report AND NDT_Instrument by
Instrument# = Number

Certificate AND Tests by
Certi_No. = Certi_No.
Batch = Batch

Certificate AND Forging_Definition by
ID_No. = ID_No.
Ver. = Ver.

New_Manuf._Plan AND RMC-Scrap by
RMC = RMC Code

New_Manuf._Plan AND Operation by
MP_No. = MP_No.
Issue = Issue

NDT Report AND Op._File by
MP_No. = MP_No.
Issue = Issue
Op_No. = No.

Steelord AND Ingot_Requirement by
ST_No. = ST_No.
Version = Version

Tests AND Specimen by
Specimen1 = Specimen
Code = Code

Tests AND Specimen by
Specimen2 = Specimen
Code = Code

Tests AND Specimen by
Specimen3 = Specimen
Code = Code

New_Manuf._Plan AND New_Test_Unit by
MP No. = MP No.
Issue = Issue

Manuf._Plan AND Test_Unit by
MP No. = MP No.
Issue = Issue

Spec._Index AND Chem._Range by
FES = FES

Spec._Index AND Mech._Property by
FES = FES
Group = Group

Spec._Creation AND Chem._Range by
Source = Source
Spec = Specification

Spec._Creation AND Chem._Range by
Source = Source
Spec = Specification
Grade = Grade

Spec._Creation AND Chem._Range by
Source = Source
Spec = Specification
GI = GI

Spec._Creation AND Spec._Index by
Group = Group

Convert_Spec. AND Spec._Index by
GroupA = Group

Convert_Spec. AND Chem_Range by
FES = FES

Convert_Spec. AND Spec._Index by
FES = FES
Group = Group

Modify_Spec. AND Spec._Index by
FES = FES
Group = Group

Modify_Spec. AND Chem._Range by
FES = FES

Modify_Spec. AND Mech._Property by
FES = FES
Group = Group
No. = No.

Modify_Spec. AND Mech_Property by
FES = FES
Group = Group
No.2 = No.

Modify_Spec. AND Mech._Property by
FES = FES
Group = Group
No.3 = No.

Modify_Spec. AND Mech._Property by
FES = FES
Group = Group
No.4 = No.

Spec._Index AND Forging_Definition by
FES = FES
Group = Group

Chem._Range AND Steelord by
FES = FES

Chem._Range AND Customer by
Source = Customer

Analysis AND Analysis_Index by
Cast = Cast

Test_Certificate AND SO_Status by
F_No. = FEL Order No.

Test_Certificate AND Certificate by
Certi_No. = Certi_No.
Batch = Batch

Check_Analysis AND Analysis_Index by
Cast = Cast

FSLladle AND Analysis_Index by
Cast_No. = Cast

Docu._Matrix AND Distribution by
Document = Document
Docu_No. = Docu_No.

Docu._Matrix AND Issue_Status by
Document = document
Docu_No. = Docu_No.

Docu._Sent AND Approve_Issue by
Document = Document
Docu_No. = Docu_No.

Spec._Creation AND Customer by
Source = Customer

Print U/S Report AND U/S_Report by
NDT No. = NDT_No.
Op_No. = Op_No.
Batch_No. = Batch

Print MPI Report AND MPI_Report by
NDT No. = NDT_No
Op_No. = Op_No.
Batch_No. = Batch

Print Dye Pen AND Dye_Report by
NDT No. = NDT_No.
Op_No. = Op_No.
Batch_No. = Batch

Print Visual AND Visual_Report by
NDT No. = NDT_No.
Op_No. = Op_No.
Batch_No. = Batch

Print Sulphur AND Sulphur_Report by
NDT No. = NDT_No.
Op_No. = Op_No.
Batch_No. = Batch

Print_Cert. AND Certificate by
Cert._No. = Certi_No.
Batch = Batch

Repeatsteel AND Issued_Steelord by
Repeat_Steel _No. = ST_No.
Issue = Version

Melting Range AND Forging_Definition by
ID_No. = ID_No.
Ver. = Ver.

Modify_Spec. AND Customer by
Source = Customer

Modify_Spec. AND Spec._Index by
FES = FES

Modifymanuf AND MP_Status by
Group_No. = FEL Order No.
Ver. = Ver.

IDN_Report AND IDN_Item by
No. = No.

Forging_Definition AND IDN_Item by
ID_No. = ID_No.
Ver. = Ver.

Documents AND Docu._Depart
Document = Document

Docu._Distribute AND Docu._Matrix by
Document = Document
Docu_No. = Docu_No.

CDSL AND CDSL_Table by
CDSL = CDSL
Issue = Issue

Enter_An_IDN AND IDN_Report by
No. = No.

Enter_a_Concession AND Concession by
Con_No. = Con_No.

Docu._Approval AND Docu._Sent by
Document = Document
Docu_No. = Docu_No.

Qp_Datafile AND Qp_OP_Datafile by
QP_No. = QP_No.
Issue = Issue

MT_Spec. AND Chem._Range by
FES = FES

MT_Spec. AND Mech._Property by
FES = FES
Group = Group

MT_Spec. AND MT_Datafile by
MT_No. = MT_No.
Issue = Issue

SP_Datafile AND SP_Comments by
SP_No. = SP_No.
Issue = Issue

FP_Datafile AND FP_Comment by
FP_No. = FP_No.
Issue = Issue
Batch = Batch

CPP AND CPP_Table by
CPP = CPP
Issue = Issue

Modifysteel AND Issued_Steelord by
Steel No. = ST_No.
Version = Version

Modifymanuf AND MP_Status by
MP_No.1 = MP No.
Issue1 = Issue

Chen._Range AND Forging_Definition by
FES = FES

Issue_MP_in_Computer AND Manuf. Plan by
Manuf. Plan = MP No.
Issue = Issue

Print_Steel_Order AND SO_Status by
ST_No. = ST_No.
Version = Version

Print_Steel_Order AND SO_Status by
FEL Order No. = FEL Order No.

Print_Steel_Order AND SO_Status by
ST_No. = ST_No.
Version = Version

Manufacturing AND Status by
F_No. = FEL Order No.

Print IDN Report AND IDN_Report by
IDN No. = No.

Print Concession AND Concession by
Concession No. = Con_No.

Printcard AND Manuf. Plan by
MP_No. = MP No.
Isuue = Issue

Status AND Forging_Definition by
ID_No. = ID_No.
Ver. = Ver.

Forging_ID._Creation AND Customer by
Customer = Customer

Forging_ID._Creation AND Chem._Range by
FES = FES

Forging_ID._Creation AND Spec._Index by
FES = FES
Group = Group

Forging_ID._Creation AND Forging_Definition by
Ver = Ver

Status AND Forging_ID._Creation by
FEL Order No. = F_No.

Modify_Forge_ID. AND Forging_Definition by
ID_No. = ID_No.
Ver. = Ver.
A = Status

Modify_Forge_ID. AND Customer by
Customer = Customer

Modify_Forge_ID. AND Chem._Range by
FES = FES

Modify_Forge_ID. AND Spec._Index by
FES = FES
Group = Group

Forging_ID._Issue AND Forging_Definition by
ID_No. = ID_No.
Ver. = Ver
A = Status

Forging_ID._Issue AND Customer by
Customer = Customer

Forging_ID._Issue AND Chem._Range by
FES = FES

Forging_ID._Issue AND Spec._Index by
FES = FES
Group = Group

Forging_ID._Issue AND Forging_Definition by
ID_No. = ID_No.

Convert_Spec. AND Mech._Property by
FES = FES
Group = Group
No. = No.

Convert_Spec. AND Mech._Property by
FES = FES
Group = Group
No.2 = No.

Convert_Spec. AND Mech._Property by
FES = FES
Group = Group
No.3 = No.

Convert_Spec. AND Mech._Property by
FES = FES
Group = Group
No.4 = No.

Modify_Melting_Range AND Spec._Index by
FES = FES
Group = Group

Modify_Melting_Range AND FEL Range by
M_Melting No. = Melting No.

Modify_Melting_Range AND Issued_Steelord by
M_Melting No. = Melting No.

Design_Melt.Range AND Matching_Melt_Range by
FEL Order No. = FEL Order No.

Design_Melt.Range AND Spec._Index by
FES = FES
Group = Group

Remanufacuring AND Status by
F_No. = FEL Order No.

Repeatsteel AND Status by
F_No. = FEL Order No.

Spec._Creation AND Chem._Range by
Source = Source
Spec = Specification
PreGrade = Grade

Manufacturing AND Status by
L_No. = FEL Order No.

Melting Range AND Status by
L_No. = FEL Order No.

Melting Range AND Status by
FEL Order 1 = FEL Order No.

Melting Range AND Status by
FEL Order 2 = FEL Order No.

Melting Range AND Status by
FEL Order 3 = FEL Order No.

Melting Range AND Status by
FEL Order 4 = FEL Order No.

HTP AND HTP_Table by
HTP = HTP
Srep = Step
Issue = Issue

QP._Data_Entry AND New_Manuf._Plan by
MP_No. = MP No.
Issue = Issue

Print_QP. AND QP_Datafile by
QP = QP_No.
Issue = Issue

QA._Docu. AND QA_Distribute by
Document No. = Document No.
Issue = Issue

QA._Docu. AND Amendment by
Document No. = Document No.
Issue = Issue

New_Manuf._Plan AND FP_Datafile by
MP No. = FP_No.

New_Manuf._PLan AND HTP by
MP No. = HTP

FP_Data_Entry AND New_Manuf._Paln by
FP_No. = MP No.
Issue = Issue

FP_Status AND Ingots by
ST_No. = ST_No.
Version = Version
Letter = Letter

MT_Spec. AND New_Manuf._Plan by
MT_No. = MP No.

Issued_QP AND Issued_QP_OP by
QP_No. = QP_No.
Issue = Issue

Issued_MT AND Chem_Range by
FES = FES

Issued_MT AND Mech._Property by
FES = FES
Group = group

Issued_MT AND Issued_MT_Comment by
MT_No. = MT_No.
Issue = Issue

Issued_SP AND Issued_SP_Comment by
SP_No. = SP_No.
Issue = Issue

Issued_FP AND Issued_FP_Table by
FP_No. = FP_No.
Issue = Issue
Batch = Batch

Issued_CDSL AND Issued_CDSL_Table by
CDSL = CDSL
Issue = Issue

Issued_CPP AND Issued_CPP_Table by
CPP = CPP
Issue = Issue

Issued_HTP AND Issued_HTP_Table by
HTP = HTP
Step = Step
Issue = Isssue

FP_Data_Entry AND FP_Datafile by
FP_No. = FP_No.
Issue = Issue

QP_Data_Entry AND QP_Datafile by
MP_No. = QP_No.
Issue = Issue

Print_FP AND FP_Datafile by
FP_No. = FP_No.
Issue = Issue
Batch = Batch

Standard_MPs AND Standard_OPs by
MP_No. = MP_No.
Issue = Issue

Standard_MPs AND RMC-Scrap by
RMC = RMC Code

Stand._MP_Selection AND Standard_MPs by
MP_No. = MP_No.
Issue = Issue

Issued_MP_Selection AND Manuf. Plan by
MP_No. = MP No.
Issue = Issue

Remanufactuing AND Status by
L_No. = FEL Order No.

Standard_MPs AND Standard_Test by
MP_No. = MP_No.
Issue = Issue

Print_QP. AND SP_Datafile by
QP = SP_No.
Issue = Issue

Print_QP. AND CPP by
QP = CPP
Issue = Issue

Print_QP.AND CDSL by
QP = CDSL
Issue = Issue

Print_QP. AND Issued_QP by
QP = QP_No.
Issue = Issue

Print_QP. AND Issued_MT by
QP = MT_No.
Issue = Issue

Print_QP. AND Issued_SP by
QP = SP_No.
Issue = Issue

Print_QP. AND Issued_CPP by
QP = CPP
Issue = Issue

Print_QP. AND Issued_CDSL by
QP = CDSL
Issue = Issue

Print_HTP AND HTP by
HTP_No. = HTP
Issue = Issue
Step = Step

Print_HTP AND Issued_HTP by
HTP_No. = HTP
Issue = Issue
Step = Step

Print_FP AND Issued_FP by
FP_No. = FP_No.
Issue = Issue
Batch = Batch

New_Issue_QP AND Issued_QP by
QP_No. = QP_No.
Issue = Issue

New_Issue_QP AND QP_Datafile by
QP_No. = QP_No.
N_I = Issue

New_Issue_HTP AND Issued_HTP by
HTP = HTP
Step = Step
Issiue = Issue

New_Issue_HTP AND HTP by
HTP = HTP
Step = Step
N_I = Issue

New_Issue_FP AND Issued_FP by
FP_No. = FP_No.
Ingot = Batch
Issiue = Issue

QA._Docu._Distribute AND QA._Docu by
Document No. = Document No.
Issue = Issue

Ingot_Requirement AND Ingots by
ST_No. = ST_No.
Version = Version

Ingot_Requirement AND Ingot_Status by
ST_No. = ST_No.
Version = Version

Issued_Ingot_Requir. AND Ingots by
ST_No. = ST_No.
Version = Version

Issued_Ingot_Requir. AND Ingot_Status by
ST_No. = ST_No.
Version = Version

Ingot_Types AND Ingots by
Ingots = Ingot Size

Ingot_Types AND FSLingot by
Ingots = INGOTS

Checkprogramme AND FSLproc by
Item = Item
Issue = Issue

QP_Status AND Forging_Definition by
FEL Order No. = FEL Order No.
Ver. = Ver.

Ingots AND Ingot_Status by
ST_No. = ST_No.
Version = Version
Letter = Letter

Issued_QP AND QP_Status by
QP_No. = QP_No.
Issue = Issue

FP_Datafile AND FP_Status by
FP_No. = FP_No.
Issue = Issue
Batch = Batch

HTP AND HTP_Status by
HTP = HTP
Step = Step
Issue = Issue

Issued_FP AND FP_Status by
FP_No. = FP_No.
Issue = Issue
Batch = Batch

Issued_HTP AND HTP_Status by
HTP = HTP
Issue = Issue
Step = Step

New_Manuf._Plan AND CDSL by
MP No. = CDSL

New_Manuf._Plan AND SP_Datafile by
MP No. = SP_No.

New_Manuf._Plan AND CPP by
MP No. = CPP

QP_Datafile AND QP_Status by
QP_No. = QP_No.
Issue = Issue

Manuf. Plan AND Issued_CDSL by
MP No. = CDSL

Manuf._Plan AND Issued_FP by
MP_No. = FP_No.

Manuf. Plan AND Issued_HTP by
MP No. = HTP

Manuf. Plan AND Issued_MT by
MP No. = MT_No.

Manuf. Plan AND Issued_SP by
MP No. = SP_No.

Manuf. Plan AND Issued_CPP by
MP No. = CPP

Status AND Forging_ID._Creation by
FEL Order No. = Last_No.

MP_Status AND HTP_Status by
MP No. = HTP

Forging_Definition AND MP_Status by
FEL Order No. = FEL Order No.
Ver. = Ver.

New_Manuf._Plan AND MP_Status by
MP No. = MP No.
Issue = Issue

Manuf. Plan AND MP_Status by
MP No. = MP No.
Issue = Issue

Issued_MP_in_Computer AND Status by
F_No. = FEL Order No.

Issued_MP_in_Computer AND
Forging_Definition by
ID_No. = ID_No.
Ver. = Ver.

Issued_MP_in_Computer AND MP_Status by
Group_No. = FEL Order No.
Ver. = Ver.

Issued_MP_in_Computer AND MP_Status by
MP_No.1 = MP No.
Issue1 = Issue

Modifymanuf AND Manuf. Plan by
Manuf. = MP No.
C_I = Issue

NDT Report AND Forging_Definition by
ID_No. = ID_No.
Ver. = Ver.

NDT Report AND MP_Status by
Group_No. FEL Order No.
Ver. = Ver.

Test_Certificate AND Forging_Definition by
ID_No. = ID_No.
Ver. = Ver.

Test_Certificate AND MP_Status by
Group_No. = FEL Order No.
Ver. = Ver.

Test_Certificate AND Manuf. Plan by
MP_No. = MP No.
Issue = Issue

ModifySteel AND SO_Status by
Steel No. = ST_No.

Repeatsteel AND Status by
L_No. = Status

FSLLadle AND SO_Status by
F_No. = FEL Order No.
FIP = FIP

FSLLadle AND SO_Status by
L_No. = FEL Order No.
FIP = FIP

FSLLadle AND SO_Status by
FEL Order 1= FEL Order No.
FIP = FIP

FSLLadle AND SO_Status by
FEL Order 2 = FEL Order No.
FIP = FIP
FSLLadle AND SO_Status by
FEL Order 3 = FEL Order No.
FIP = FIP

FSLLadle AND SO_Status by
FEL Order 4 = FEL Order No.
FIP = FIP

Melting Programme AND SO_Status by
F_No. = FEL Order No.

Melting Programme AND SO_Status by
L_No. = FEL Order No.

Melting Programme AND SO_Status by
FEL Order 1 = FEL Order No.

Melting Programme AND SO_Status by
FEL Order 2 = FEL Order No.

Melting Programme AND SO_Status by
FEL Order 3 = FEL Order No.

Melting Programme AND SO_Status by
FEL Order 4 = FEL Order No.

New_Issue_FP AND FP_Datafile by
FP_No. = FP_No.
Ingot = Batch
N_I = Issue

SO_Status AND Ingot_Status by
ST_No. = ST_No.
Version = Version

FSLingot AND SO_Status by
Item = Item
Issue = Issue

FSLproc AND SO_Status by
Item = Item
Issue = Issue

Steelord AND Spec._Index by
FES = FES
Group = Group

Issued_Steelord AND SO_Status by
ST_No. = ST_No.
Version = Version

Issued_Steelord AND Issued_Ingot_Requir. by
ST_No. = ST_No.
Version = Version

Issued_Steelord AND FEL Range by
Melting No = Melting No.

Issued_Steelord AND Chem._Range by
FES = FES

SO_Status AND Forging_Definition by
ID_No. = ID_No.
Ver. = Ver.

MT_Data_Entry AND MP_Status by
MT_No. = MP No.
Issue = Issue

MT_Data_Entry AND MP_Spec. by
MT_No. = MT No.
Issue = Issue

HTP_Data_Entry AND MP_Status by
HTP_No. = MP No.
Issue = Issue

HTP_Data_Entry AND HTP by
HTP_No. = HTP
Issue = Issue
Step = Step

SP_Data_Entry AND MP_Status by
SP_No. = MP No.
Issue = Issue

SP_Data_Entry AND SP_Datafile by
SP_No. = SP_No.
Issue = Issue

CPP_Data_Entry AND MP_Status by
CPP_No. = MP No.
Issue = Issue

CPP_Data_Entry AND CPP by
CPP_No. = CPP
Issue = Issue

New_Issue_MT AND Issued_MT by
MT_No. = MT_No.
Issue = Issue
Batch = Batch

New_Issue_MT AND MT_Spec. by
MT_No. = MT_No.
N_I = Issue
Batch = Batch

New_Issue_CDSL AND Issued_CDSL by
CDSL = CDSL
Issue = Issue

New_Issue_CDSL AND CDSL by
CDSL = CDSL
Issue = Issue

New_Issue_SP AND Issued_SP by
SP_No. = SP_No.
Issue = Issue

New_Issue_SP AND SP_Satafile by
SP_No. = SP_No.
Issue = Issue

New_Issue_CPP AND Issued_CPP by
CPP = CPP
Issue = Issue

New_Issue_CPP AND CPP by
CPP = CPP
N_I = Issue

SO_Status AND IDN_Item by
FEL Order No. = FirstNo.

Status AND SO_Status by
FEL Order No. = FEL Order No.

Repeat_QP AND New_Manuf._Plan by
MP_No. = MP No.
Issue = Issue

Repeat_QP AND QP_Datafile by
MP_No. = QP_No.
Issue = Issue

Repeat_QP AND Issued_QP by
MP_No. = QP_No.
Issue = Issue

Repeat_MT AND MP_status by
MT_No. = MP No.
Issue = Issue

Repeat_MT AND MT_Spec. by
MP_No. = MT_No.
Issue = Issue
Batch = Batch

Repeat_MT AND Issued_MT by
P_NT_No = MT_No.
P_Issue = Issue
Batch = Batch

Repeat_HTP AND MP_Statusd by
HTP_No. = MP No.
Issue = Issue

Repeat_HTP AND HTP by
HTP_No. = HTP
Issue = Issue
Step = Step

Repeat_HTP AND Issued_HTP by
P_HTP_No. = HTP
P_Issue = Issue
P_Step = Step

Repeat_FP AND New_Manuf._Plan by
FP_No. = MP No.
Issue = Issue

Repeat_FP AND FP_Datafile by
FP_No. = FP_No.
Issue = Issue

Repeat_FP AND Issued_FP by
P_FP_No. = FP_No.
P_Issue = Issue
P_Batch = Batch

Repeat_CPP AND QP_Status by
CPP_No. = QP_No.
Issue = Issue

Repeat_CPP AND CPP by
CPP_No. = CPP
Issue = Issue

Repeat_CPP AND Issued_CPP by
P_CPP_No. = CPP
P_Issue = Issue

Repeat_SP AND QP_Status by
SP_No. = QP_No.
Issue = Issue

Repeat_SP AND SP_Datafile by
SP_No. = QP_No.
Issue = Issue

Repeat_SP AND Issued_SP by
P_SP_No. = SP_No.
P_Issue = Issue

Generating_CDSL AND QP_Status by
CDSL_No. = QP_No.
Issue = Issue

Generating_CDSL AND CDSL by
CDSL_No. = CDSL
Issue = Issue

Print_MT AND MT_Spec. by
MT_No. = MT_No.
Issue = Issue

Print_MT AND Issued_MT by
MT_No. = MT_No.
Issue = Issue

QWON1 AND Status by
FEL Order No. = FEL Order No.

QWON1 AND MP_Status by
FEL Order No.1 = FEL Order No.
Ver. = Ver.

List_QA._Document AND QA_Distribute by
Department = Department

QWON1 AND SO_Status by
FEL Order No. = FEL Order No.

Manufacturing AND MP_Status by
WON = FEL Order No.
Ver. = Ver.

Manufacturing AND Forging_Definition by
ID_No. = ID_No.
Ver. = Ver.

Steelord AND New_Manuf._Plan by
MP_No. = MP No.
Issue = Issue

Steelord AND Manuf._Plan by
MP_No. = MP No.
Issue = Issue

Modify_Melting_Range AND Steelord by
ST_No. = ST_No.
Version = Version

New_Manuf._Plan AND QP_Datafile by
MP No. = QP_No.

Print_CPP AND CPP by
CPP = CPP
Issue = Issue

Print_CPP AND Issued_CPP by
CPP = CPP
Issue = Issue

Print_SP AND SP_Datafile by
SP = SP_No.
Issue = Issue

Print_SP AND Issued_SP by
SP = SP_No.
Issue = Issue

Print_CDSL AND CDSL by
CDSL = CDSL
Issue = Issue

Print_CDSL AND Issued_CDSL by
CDSL = CDSL
Issue = Issue

MP_Status AND MT_Status by
MP No. = MT_No.

Melting Range AND SO_Status by
F_No. = FEL Order No.

Test_Certificate AND Status by
F_No. = FEL Order No.

Ingot_Requirement AND SO_Status by
ST_No. = ST_No.
Version = Version

Issued_Ingot_Requir. AND SO_Status by
ST_No. = ST_No.
Version = Version

NDT Report AND SO_Status by
F_No. = FEL Order No.

Print U/S Report AND H_U/S_Scan by
NDT No = NDT_No.
OP_No. = OP_No.

NDE_Procedure AND NDE_Tables by
NDE_No. = NDE_No.
Issue = Issue

Create_NDE AND NDE_Procedure by
NDE_No. = NDE_No.
Issue = Issue

Print_NDE_Procedure AND NDE_Procedure by
NDE_No. = NDE_No.
Issue = Issue

Repeat_NDE AND NDE_Procedure by
P_NDE_No. = NDE_No.
P_Issue = Issue

Repeat_NDE AND NDE_Procedure by
NDE_No. = NDE_No.
Issue = Issue

Printdiscrepancy1 AND Discrepancy by
Item = Item
Issue = Issue

Analysis_Index AND SO_Status by
Cast_No. = Cast No.

List_Docu._Sent AND Docu._Sent by
FEL Order No. = FEL Order No.

Print_Manuf. AND Manuf. Plan by
Manuf. Plan = MP No.
Issue = Issue

Print_Manuf. AND Status by
F_No. = FEL Order No.

Print_Manuf. AND Forging_Definition by
ID_No. = ID_No.
Ver. = Ver.

Print_Manuf. AND MP_Status by
Group_No. = FEL Order No.
Ver. Ver.

Print_Manuf. AND MP_Status by
MP_No.1 = MP No.
Issue1 = Issue

Print_Steel_Order AND Steelord by
ST_No. = ST_No.
Version = Version

Steelord AND SO_Status by
ST_No. = ST_No.
Version = Version

APPENDIX 3

LISTING OF PROCEDURES IN INDIVIDUAL MENUS

3-1 Procedures in Metallurgical Functions

Menu Title	Procedure	Menu Name
Technical Planning		
	Create_Forging_ID.	Create a New Forging ID.
	Edit_Forging_ID.Edit an Unissued Forging ID.	
	Forging_ID._N_Issue	Revision of a Forging ID.
	View Forging ID.	View the Forging ID. Datafile
	QWON1	Forging Status Query
Manufacturing Plan		
	Edit_Stand._MPs	View Standard Manuf. Plan
	Issue_MP_in_Computer	Make Manuf. Plan Issued
	Manufacturing	Start a New Manuf. Plan
	Create_Manuf	Create a New Manuf. Plan
	Modify_Manuf	Edit Current Manuf. Plan
	Print_Manuf	Print Manuf. Plan Report
	Remanufacturing	Repeat Proforma Manuf. Plan
	ViewMP	View Issued Manuf. Plan
Steel Order		
	Melting Range	Start Steel Order
	Modifysteel	Create a New Issue of SO
	Modify_Melt.Range	Modify FEL Melt.Range in Current Steel Order
	Edify_St	Edit Current Steel Order Except Melt.Range
	Print_Steel_Order	Print a Steel Order Report
	Repeatsteel	Repeat a Previous Steel Order
	ViewSt	View Issued Steel Order
Forging Procedure		
	FP_Data_Entry	Create a Forging Procedure
	Edit_FP_Datafile	Edit None Issued FP.
	New_Issue_FP	Create a New Issue of FP.
	Print_FP	Print a Forging Procedure
	Repeat_FP	Repeat Previous FP.
Quality Plan Operation		
	CPP_Data_Entry	Create a CPP Procedure
	FP_Data_Entry	Create a Forging Procedure
	HTP_Data_Entry	Create a HT. Procedure
	MT_Data_Entry	Create a MT. Procedure
	QP._Data_Entry	Create a QP._Operation
	SP_Data_Entry	Create a Special Procedure

None Issued QP. Edition		
	Edit_CDSL	Edit QP_CDSL Data File
	Edit_CPP	Edit QP_CPP Data File
	Edit_FP_Datafile	Edit QP_FP Data File
	Edit_HTP	Edit QP_HTP Data File
	Edit_MT_Spec.	Edit QP_MT Data File
	Edit_SP_Datafile	Edit QP_SP Data File
	Edit_QP_OP_Datafile	Edit QP_Operation Data File
New Issue QP. Creation		
	New_Issue_CDSL	New Issue of CDSL
	New_Issue_CPP	New Issue of CPP
	New_Issue_FP	New Issue of FP
	New_Issue_HTP	New Issue of HTP
	New_Issue_MT	New Issue of MT
	New_Issue_SP	New Issue of SP
	New_Issue_QP	New Issue of QP
Quality Plan Printing		
	Print_CDSL	Print a CDSL Procedure
	Print_CPP	Print a CPP Procedure
	Print_FP	Print a Forging Procedure
	Print_HTP	Print a HT. Procedure
	Print_MT	Print a MT. Procedure
	Print_SP	Print a Special Procedure
	Print_QP	Print a Quality Plan
Repeat Previous QP		
	Generating_CDSL	Create a New CDSL
	Repeat_CPP	Repeat Previous CPP.
	Repeat_FP	Repeat Previous FP.
	Repeat_HTP	Repeat Previous HT.
	Repeat_MT	Repeat Previous MT.
	Repeat_SP	Repeat Previous SP.
	Repeat_QP	Repeat Previous QP_OP.
Ingot Requirement		
	Modify_Ingot	Design/Edit Current Ingot Requirement Datafile
	Print_Steel_Order	Print Steel Order Report
	View_Ingot_Requi.	View Issued Ingot Requirement Datafile

3-2 Procedures in Quality Assurance Functions

Menu Title	Procedure	Menu Name
Knowledge Data Bank		
	Modify_Spec.	Edit Uncirculated Spec.
	Spec._Creation	Create a Specification
	Chemitospec	Create a New Version of Spec.
	Data Entry Menu	Withdrawal of Specification

Document Distribution		
	Docu._Approval	Enter a Record for Document Approval from Customer
	Docu._Distribution	Enter a Record for internal Document Distribution
	Delete_None_Distr.	Delete None Distributed Records
Listing Docu. Distributions		
	List_Docu._Approval	Documents Awaiting Customer Approval
	List_Docu._Reissue	Rejected Document Awaiting Re-Issue
	List_Docu._Sent	Documents Sent to Customers
	Print_NRD_Report	List Internal Distributions Not yet returned
	List_QA._Document	List Document Distribution for Specified Department
FSL Melting Programmes		
	Checkprogramme	Check FSL Melting Programme
	Melting Programme	Melting programme Data-Entry
	Pintdiscrepancy	Print Discrepancy Report
Chemical Analysis		
	Check_Analysis	Check Forging Cast Analysis
	FSLLadle	Forging Analysis Data-Entry
	Printcard	Print Prod. Traveller Cards

3-3 Procedures in Test Results Functions

Menu Title	Procedure	Menu Name
Test Certificates		
	Check_Analysis	Check Cast Analysis
	Print_Cert.	Test Certificate Printing
	Test_Certificate	Test Certificate Design Data Entry
	View_Certificate	View/Edit Test Certificate
NDT Report Menu		
	NDT Report	Start a NDT Report (first time)
View/Edit NDT Reports		
	View_U/S	View/Edit Ultrasonic Report
	View_MPI	View/Edit M. Particle Report
	View_Dye	View/Edit Dye Pen. Report
	View_Visual	View/Edit Visual Report
	View_Sulphur	View/Edit S. Print Report
Probe/Instrument/Block		
	U/S_Probe	Probe File
	NDT_Instrument	NDT Instrument
	Block_Listing	Ultrasonic Block Selection

NDT Report Printing

	Print U/S Report	Print U/S Test Report
	Print MPI Report	Print MPI Report
	Print Dye Pet.	Print Dye Pen. Report
	Print Visual	Print Visual Report
	Print Sulphur	Print S. Print Report

NDT Procedure

	Create_NDE	NDE Procedure Data-Entry
	Edit_NDE	Edit a NDE Procedure
	Print_NDE_Procedure	Print a NDE Procedure
	Repeate_NDE	New Issue of NDE Procedure

APPENDIX 4

DESCRIPTIONS OF QUERY PROCEDURES

4-1 Procedures in Metallurgical Functions

4-1-1 Technical Planning Menu

4-1-1-1 Create a New Forging ID.

Procedure 1: *Create_Forging_ID.*

Functions: This procedure is for creating a Forging Definition record. The procedure can create a temporary file, Temp_ID., in which user's data entry can be checked to ensure they comply with system rules before physically recorded them in Forging Definition file.

```
DQL QUERY
define temp"Found"Text 1.  define temp"ID_No."text 4.
define temp"Group"Text 1.  define temp"F_Order"Numeric String 7.
define temp"L_Order"Numeric String 7.  define temp"Order"Numeric String 7.
while current status not=1 do
message"Please enter the FEL Order No. first.".
input using Forging_ID._Creation into "Temp_ID".
if current status =1 then
exit
end
if current status =2 then
if Temp_ID Are you sure?=no then
message"You should press ESC to exit. Exit now."window.
exit
end
assign temp ID_No.:=Temp_ID ID_No..
assign temp F_Order:=firstc(temp_ID FEL Order No.,7).
assign temp Order:=temp F_Order.
if temp_ID FES not = blank then
assign temp Group:=Temp_Id Group.
end
if Temp_ID Last_No. not = blank then
assign temp L_Order:=Temp_ID Last_No..
if temp F_Order > temp L_Order then
message"Wrong Sequence of FEL Order No.(s) specified, Exit."window.
exit
end  end
for Identification with ID_No.=temp ID_No. and Ver.="A";
assign temp Found:="Y".
end
if temp Found="Y" then
assign temp ID_No.:=highest of Identification with (Ver.="A")ID_No.+1;
assign temp ID_No.:=jointext(firstc("000",4-length(temp ID_No.)),temp ID_No.).
end
enter a record in Identification
copy all from Temp_ID;
ID_No.:=temp ID_No.;  group:=temp Group.
if temp L_Order not=blank then
while temp Order not=temp L_Order+1 do
```

```
enter a record in status
copy all from Temp_ID;
ID_No. :=temp ID_No.;  FEL Order No.:=temp Order.
assign temp Order:=temp Order+1.
end
else
enter a record in status
copy all from Temp_ID;
ID_No.:=temp ID_No.;  FEL Order No.:=temp F_Order.
end  end  exit
End
```

4-1-1-2 Edit an Un-issued Forging ID.

Procedure 1: ***Edit_Forging_ID.***

Functions: This procedure is designed to allow user to access and edit current Forging Definition records. It can check user's data entry to identify certain key information which may violate system integrity rules. For example, the modification to the FEL Order No. bundle must comply with that new bundle is within previous range and this requires to create a new version of the Forging Definition record.

```
DQL QUERY
define temp "ID_No."Numeric String 4.  define temp "Group"Text 1.
define temp "F_Order"Numeric String 7.  define temp "L_Order"Numeric String 7.
define temp "Order"Numeric String 7.
while current status not=1 do
message"Press CTRL-F10 to select one to modify.".
input using Modify_Forging_ID. into "Temp_ID".
if current status=1 then
exit
end
if current status=2 then
if Temp_ID Are you sure? not = Yes then
message"You should press ESC to exit. Exit now."window.
exit end
assign temp ID_No.=Temp_ID ID_No..
assign temp F_Order:=firstc(temp_ID FEL Order No.,7).
assign temp Order:=temp F_Order.
if temp_ID FES not = blank then
assign temp Group:=Temp_ID Group.
end
if Temp_ID Last_No. not = blank then
assign temp L_Order:=Temp_Id Last_No..
if Temp F_Order > temp L_Order then
message"Wrong Sequence of FEL Order No.(s) specified, Exit."window.
exit
end  end
if Temp_ID Status=no then
for Identification with(ID_No.=Temp_ID ID_No. and Ver.=Temp_ID Ver.);
modify records
copy all from Temp_ID;
Initial := current user name; Date := current date;
FES:=Temp_ID FES; group:=temp Group.
exit  end  end  end
End.
```

4-1-1-3 Revision of a Forging ID.

Procedure 1: ***Forging_ID._N_Issue***

Functions: This procedure is for creating a new issue of Forging Definition record. The procedure can create a temporary file, "Temp_ID", in which user's data entry can be checked.

```
DQL QUERY
define temp "Ver_Name"Text 1. define temp "F_Order"Numeric String 7.
define temp "L_Order"Numeric String 7. define temp "Order"Numeric String 7.
while current status not=1 do
message"CTRL-F10 to select one for new issue.".
input using Forging_ID._Issue into "Temp_ID".
if current status =1 then
exit
end
if current status=2 then
if temp_ID Are you sure? not=Yes then
message"You should press ESC to exit. Exit now."window.
exit
end
assign temp F_Order:=firstc(temp_ID FEL Order No.,7)
assign temp Order:=temp F_Order. assign temp Ver_Name := Temp_ID N_Issue2.
if Temp_ID FES not = blank then
assign temp Group:=Temp_ID Group.
end
if Temp_ID Last_No. not = blank then
assign temp L_Order:=Temp_ID Last_No..
end
if Temp_ID Status=Yes then
enter a record in Identification
copy all from Temp_ID;
Status:=no; Initial :=current user name.
group:=temp group; Ver.:=temp Ver_Name.
if temp L_Order not=blank then
while temp Order not=temp L_Order+1 do
modify records in Status with(FEL Order No.=temp Order)
Ver.:=temp Ver_Name. assign temp Order:=temp Order+1.
end
else
modify records in Status with(FEL Order No.=temp F_Order)
Ver.:=temp Ver_Name.
end end end exit
End.
```

4-1-1-4 View the Forging ID Datafile

(Through Data Entry Menu)

4-1-1-5 Forging Status Query

Procedure 1: ***QWON1***

Functions: This procedure is designed to question the database that for a given FEL Order Number, what is the key identifiers in some data files such as Forging Definition, Manufacturing Plan, Steel Order, Cast Analysis and so on. This is achieved through the procedure relationships with these files by lookup function defined in individual fields in the procedure data entry form. Followings are these fields definitions:-

1. FEL Order No.	Num. String 7.
2. Text1	If (ID_No.=blank, blank, "Forging Def. ID. is").
3. ID_No.	Lookup Status "ID_No.".
4. Ver.	Lookup Status "Ver.".
5. FEL Order No.1	Lookup Status "FEL Order No.".
6. Text2	If (MP_No.=blank, blank, "Manufacturing Plan:").
7. MP_No.	Lookup MP_Status "MP_No.".
8. MP_I	Lookup MP_Status "Issue".
9. Text3	If (SO_No.-blank, blank, "Steel Order No. is:").
10. SO_No.	Lookup SO_Status ST_No.
11. Version	Lookup SO_Status Version.
12. Text4	If (Cast_No. = blank, blank, "Cast No. and Letter:").
13. Cast_No.	Jointext (lookup SO_Status "Cast_No.", lookup SO_Status "Letter")
14. Text5	If (FEL Order No. = blank, blank, "Press ESC to calling Menu").

4-1-2 Manufacturing Plan Menu

4-1-2-1 View Standard Manuf. Plan

Procedure 1: *Edit_Stand._MPs*

Functions: This procedure is similar to procedure 3 listed in section 4-1-2-3 except that it is authorised to senior metallurgists for accessing and editing standard manufacturing plans from which new plans can be generated by other designers.

4-1-2-2 Make Manuf. Plan Issued

Procedure 1: *Issue_MP_in_Computer*

Functions: This is a command procedure for issuing a Manuf. Plan in the computer by calling a processing procedure "Issue_MP_Data". It can check user's data entry and inform the user whether or not the information entered is correct.

```
DQL QUERY
define global"MP_No."Text 5. define global"Issue"Text 1.
define global"Imp"text 3. define global MP_No.:=data-entry Manuf. Plan.
define global Issue:=data-entry Issue. define global Imp:=data-entry O_I.
if data-entry Are you sure? not=Yes then
exit
else
if data-entry O_T = Yes then
run procedure "Issue_MP_Data".
else
message"Try again answer Yes for official issue."window.
end
end
```

Procedure 2: *Issue_MP_Data*

Functions: This procedure is for transferring Manuf. Plan data from a current file, New_Manuf. Plan, into an issued data file, Manuf. Plan, which therefore can be accessed by other users in other process planning areas or functions.

```
DQL QUERY
define temp "Seq." Number.  define temp "No." Number.
define global "MP_No." Text 5.  define global "Issue" Text 1.
define global "Imp" Text 3.
message "Please wait for DataEase Processing."
if global Issue>="B" then
 for MP_Status with (MP No.=global MP_No. and Issue<global Issue);
 enter a record in H_MP_Status.
 copy all from MP Status.
 delete records
 break
 end
end
for New_Manuf._Plan with (MP No.=global MP_No. and Issue=global Issue);
 for MP_Status
 modify records
 FIP:=Yes.
 for Identification
 modify records
 Status:=Yes.
 for Spec._Index
 modify records
 alive:=Yes.
 end
 end
end
enter a record in Manuf. plan
copy all from New_Manuf._paln.
assign temp Seq.:=1.
assign temp No. :=1.
while temp Seq.=temp No. do
for Operation with (MP No.=global MP_No. and Issue=global Issue and Seq.=temp Seq.)
 enter a record in Op._File
 copy all from Operation.
 delete records
 assign temp Seq.:=temp Seq.+1.
end
 assign temp No.:=temp No.+1.
end
for New_Test_Unit
 enter a record in Test _Unit
 copy all from New_Test_Unit.
 delete records
end
delete records
End.
```

4-1-2-3 Start a New Manuf. Plan

Procedure 1: ***Manufacturing***

Functions: This is a command procedure for starting a new Manufacturing Plan. It can check user's data entry to avoid creating duplicated plans and help user to establish links between the Manufacturing Plan and other files within the database. It runs several processing procedures and finally lets user to access and edit data in Manufacturing Plan data file by calling procedure, Modify_Mp.

```
DQL QUERY
define global "F_No."Numeric String 7. define global "L_No."Numeric String 7.
define global "New MP_No."Text 5. define global "New Issue"Text 1.
define global "Exit"Text 3. assign global F_No.:=data-entry F_No..
if data-entry L_No.=blank then
assign global L_No.:=data-entry F_No..
else
assign global L_No.:=data-entry L_No..
end
assign global New Issue:="A".
if data-entry are you Sure? not=Yes then
exit
else
run procedure "Basedata1".
run procedure "Modify_Mp".
End.
```

Procedure 2: ***Basedata1***

Functions: This is a processing procedure. It can create a Manuf. Plan or a Steel Order with basic information such as document identities, the creation date, the designer's name and so on.

```
DQL QUERY
define global "FEL Order No."Numeric String 11. define temp "ID"Text 5.
define global "F_No."Numeric String 7. define global "L_No."Numeric String 7.
assign global "FEL Order 1"Numeric String 7. assign global "FEL Order 2"Numeric String 7.
assign global "FEL Order 3"Numeric String 7. assign global "FEL Order 4"Numeric String 7.
assign global "New MP_No."Text 5. define global "New Issue"Text 1.
define global "New ST_No."Text 6. define global "Melting_No."Text 5.
define global "FES"Numeric String 4. define global "Group"Text 1.
if global Melting No.=blank then
assign global New MP_No.:=highest of MP_Status MP No.+1.
assign global New MP_No.:=jointext(firstc("000",(4-length(global New MP_No.))),
                          global New MP_No.).
enter a record in New_Manuf._plan
MP No.:=global MP_No.; Issue:=upper(global New Issue);
Date_Crd:=current date; Date:=current date;
Initial:=current user name.
for Status with FEL Order No. between global F_No. to global L_No.;
if temp ID not=ID_No. then
assign temp ID :=ID_No..
for Identification with(ID_No.=temp ID)
enter a record in MP_Status
copy all from Identification;
MP No.:=global New MP_No.; Issue:=global New Issue.
end
end
end
for Renumber_Op. with MP No.=Mast.";
enter a record in Operation
copy all from Renumber_Op.;
MP No.:=global New MP_No.; Issue:=upper(global New Issue).
```

```
 end
else
  assign global New ST_No.:=highest of SO_Status ST_No..
  assign global New ST_No.:=global New ST_No.+1.
  assign global New ST_No.:=jointext(firstc("00000",(6-length(global New ST_No.))),
                          global New ST_No.).
  if global New ST_No.="000001" then
  exit
  end
 enter a record in Steelord
 ST_No.:=global New ST_No.;  Version:=upper(global New Issue);
 Date:=current date;  Initial:=current user name;
 Approve:=no;  FES:=global FES;
 group:=global Group;  Melting No.:=global Melting No..
 enter a record in Ingot_Requirement
 ST_No.:=global New ST_No.;  Version:=upper(global New Issue);
 Approve:=no.
 for Status with FEL Order No. between global F_No. to global L_No. or
   FEL Order No.=global FEL Order 1 or FEL Order No. =global FEL Order 2 or
   FEL Order No.=global FEL Order 3 or FEL Order No. =global FEL Order 4;
 enter a record in SO_Status
 copy all from Status;
 ST_No.:=global New ST_No.;  Version:=global New Issue.
 end
End.
```

Procedure 3: *Modify_Mp*

Functions: This procedure is designed for user to access and edit current Manuf. Plans. It can re-arrange Manuf. Plan operation sequences as specified by the designer.

```
DQL QUERY
define temp "N" Number.  define temp "LSN" Number.
define temp "CN" Number.  define temp "OPN" Number.
define temp "TN" Number.  define temp "F3"text 1.
while current status not=1 do
assign temp N:=blank.  assign temp LSN:=blank.
assign temp CN:=blank.  assign temp TN:=blank.
if global F3="Y" then
message "Press SHFT-F3 to load your Manuf. Plan.".
else
message"Search using SHFT-F3, SHFT-F1 or ALT-F5.".
end
input using New_Manuf._Plan into "Temp_Mp".
if current status=1 then
exit
else
if current status=3 or current status=2 then
message "Wait for DataEase processing please...".
for New_Manuf._Plan with(M_ No.=Temp_Mp MP No. and Issue=Temp_Mp Issue);
modify records
copy all from Temp_Mp;
Date:=current date;  Initial:=current user name.
for Operation
if Seq.=blank then
if No.=blank then
```

```
if Comment=blank then
delete records
end end end end
for New_Test_Unit
if Test_No.=blank and Tests=blank then
delete records
end
if current status=2 then
for Operation
assign temp TN:=temp TN+1.
enter a record in Renumber_Op.
copy all from Operation.
delete records
end
assign temp TN:=temp TN+2.
while temp N not=temp TN do
for Renumber_Op. with(MP No.=Temp_Mp MP No. and Issue=Temp_Mp Issue and Seq.=temp N)
assign temp LSN:=temp LSN+1.
if Operation not=blank then
assign temp CN:=temp CN+1. assign temp OPN:=temp CN.
else
assign temp OPN:=blank.
end
enter a record in Operation
copy all from Renumber_Op.;
Seq.:=temp LSN; No.:=temp OPN.
delete records
end
assign temp N:=temp N+1.
end end end end
exit
End.
```

4-1-2-4 Create a New Manuf. Plan

Procedure 1: ***Create_Manuf***

Functions: This is a command procedure that controls process for generating a new issue of Manuf. Plan reports. It can check user's data entry to prevent any error in this process and generate the next issue letter automatically, and finally, it lets user to access and edit the new Manuf. Plan.

```
DQL QUERY
define global "New MP_No."Text 5. define global "New Issue"Text 1.
define global "MP_No."Text 5. define global "Issue"Text 1.
define global "Exit"Text 3. define global "Number"Number.
if data-entry Are you sure? = Yes then
assign global MP_No. :=data-entry Manuf..
assign global New MP_No. :=global MP_No..
assign global New Issue:=data-entry New Issue.
assign global Issue:=data-entry C_I.
run procedure "MP_Creation1".
if current status =1 or global Exit ="Yes" then
exit
else
```

```
run procedure "Modify_Mp".
end
End.
```

Procedure 2: *MP_Creation1*
(Same as procedure 3 listed in section 4-1-2-7.)

Procedure 3: *Modify_Mp*
(Same as procedure 3 listed in section 4-1-2-3.)

4-1-2-5 Edit Current Manuf. Plan

Procedure 1: *Modify_Mp*
(Same as procedure 3 listed in section 4-1-2-3.)

4-1-2-6 Print Manuf. Plan Report

Procedure 1: *Print_Manuf*

Functions: This is a command procedure for printing Manuf. Plan Reports. It can print manufacturing plan reports from current Manuf. Plan file or from issued Manuf. Plan file by executing different processing procedures depending on user's request. Before it prints the reports, it collects information from certain files and assigns them to global variables which van be passed from procedure to procedure in order to achieve required format of the reports. It can also issue manufacturing plans in the computer if it is requested.

```
DQL QUERY
define global "QA1"Text 18.  define global "QA2"Text 18.
define global "QA3"Text 18.  define global "QA4"Text 18.
define global "Unit1"Text 18.  define global "Unit2"Text 18.
define global "Unit3"Text 18.  define global "Unit4"Text 18.
define global "Unit5"Text 18.  define global "Unit6"Text 18.
define global "Unit7"Text 18.  define global "Unit8"Text 18.
define global "Unit9"Text 18.  define global "Test1"Text 20.
define global "Test2"Text 20.  define global "Test3"Text 20.
define global "Test4"Text 20.  define global "Test5"Text 20.
define global "Test6"Text 20.  define global "Test7"Text 20.
define global "Test8"Text 20.  define global "Test9"Text 20.
define global "Spec."Text 50.  define global "MP_No."Text 5.
define global "Issue"Text 1.  define global "Imp"text 3.
define global "N_Op"Number.  assign global MP_No.:=data-entry Manuf. Plan.
assign global Issue :=data-entry Issue.  assign global Imp :=data-entry O_I.
if data-entry Are you sure? not=Yes then
exit
else
if data-entry O_I2=no and data-entry O_I=Yes then
run procedure "Issue_MP_Data".
end
run procedure "Assign_MP_Data".
if data-entry O_I=no then
 if global N_Op<=13 then
 run procedure "Print_MP_Format2".
 else
 if global N_Op>=35 then
 run procedure "Print_MP_Format3".
```

```
 run procedure "Print_MP_Comment".
 else
 run procedure "Print_MP_Format1".
 end end
else
 run procedure "Print_Issue_Manuf.".
end
End.
```

Procedure 2: ***Issue_MP_Data***
(Same as procedure 2 listed in section 4-1-2-2.)

Procedure 3: ***Assign_MP_Data***

Functions: This procedure is designed to generate information from database before the printing procedures are called. The information will be QA Specifications., test units and per unit requirements. This is for achieving required format of the reports.

DQL QUERY

```
define temp "Seq."Number. define temp "No."Number. define global "Spec."text 50.
define global "MP_No."Text 5. define global "Issue"Text 1.
define global "Imp"text 3. define global "QA1"Text 18.
define global "QA2"Text 18. define global "QA3"Text 18.
define global "QA4"Text 18. define global "Test1"Text 20.
define global "Test2"Text 20. define global "Test3"Text 20.
define global "Test4"Text 20. define global "Test5"Text 20.
define global "Test6"Text 20. define global "Test7"Text 20.
define global "Test8"Text 20. define global "Test9"Text 20.
define global "Unit1"Text 18. define global "Unit2"Text 18.
define global "Unit3"Text 18. define global "Unit4"Text 18.
define global "Unit5"Text 18. define global "Unit6"Text 18.
define global "Unit7"Text 18. define global "Unit8"Text 18.
define global "Unit9"Text 18. define global "N_Op"Number.
if global Imp="no" then
for New_Manuf._Plan with(MP No.=global MP_No. and Issue=global Issue);
assign temp No.:=temp No.+1. assign QA1:=jointext(temp No.,"."),Assur1).
if Assur2 not=blank then
assign temp No.:=temp No.+1. assign QA2:=jointext(temp No.,"."),Assur2).
end
if Assur3 not=blank then
assign temp No.:=temp No.+1. assign QA3:=jointext(temp No.,"."),Assur3).
end
if other not=blank then
assign temp No.:=temp No.+1. assign QA4:=jointext(temp No.,"."),other).
end
assign temp No.:=blank.
for Operation
assign global N_Op:=global N_Op+1.
end
for New_Test_Unit
case(Test_No.)
value (1): assign global Test1:=jointext("1.",Test). assign global Unit1:=Unit.
value (2): assign global Test2:=jointext("2.",Test). assign global Unit2:=Unit.
value (3): assign global Test3:=jointext("3.",Test). assign global Unit3:=Unit.
value (4): assign global Test4:=jointext("4.",Test). assign global Unit4:=Unit.
value (5): assign global Test5:=jointext("5.",Test). assign global Unit5:=Unit.
```

```
value (6): assign global Test6:=jointext("6.",Test). assign global Unit6:=Unit.
value (7): assign global Test7:=jointext("7.",Test). assign global Unit7:=Unit.
value (8): assign global Test8:=jointext("8.",Test). assign global Unit8:=Unit.
value (9): assign global Test9:=jointext("9.",Test). assign global Unit9:=Unit.
end end end
else
for Manuf. Plan with MP No.=global MP_No. and Issue=global Issue;
assign temp No.:=temp No.+1. assign QA1:=jointext(temp No.,"."),Assur1).
if Assur2 not=blank then
assign temp No.:=temp No.+1. assign QA2:=jointext(temp No.,"."),Assur2).
end
if Assur3 not=blank then
assign temp No.:=temp No.+1. assign QA3:=jointext(temp No.,"."),Assur3).
end
if other not=blank then
assign temp No.:=temp No.+1. assign QA4:=jointext(temp No.,"."),other).
end
for Op._File
assign global N_Op:=global N_Op+1.
end
for Test_Unit
case(Test_No.)
value (1): assign global Test1:=jointext("1.",Test). assign global Unit1:=Unit.
value (2): assign global Test2:=jointext("2.",Test). assign global Unit2:=Unit.
value (3): assign global Test3:=jointext("3.",Test). assign global Unit3:=Unit.
value (4): assign global Test4:=jointext("4.",Test). assign global Unit4:=Unit.
value (5): assign global Test5:=jointext("5.",Test). assign global Unit5:=Unit.
value (6): assign global Test6:=jointext("6.",Test). assign global Unit6:=Unit.
value (7): assign global Test7:=jointext("7.",Test). assign global Unit7:=Unit.
value (8): assign global Test8:=jointext("8.",Test). assign global Unit8:=Unit.
value (9): assign global Test9:=jointext("9.",Test). assign global Unit9:=Unit.
end end end end
for MP_Status with MP No.=global MP_No. and Issue=global Issue;
for Identification
for Chem.Range
assign global Spec.:=Specification.
if Grade not="NS" and Issue No. not="NS" then
assign global Spec.:=jointext(jointext(jointext(jointext(Specification, " "),Grade)," "),Issue No.).
else
if Grade not="NS" then
assign global Spec.:=jointext(jointext(Specification," "),Grade).
end
if Issue No. not="NS" then
assign global Spec.:=jointext(jointext(global Spec.,"Issue:"),Issue No.).
end end end
for Spec._Index
if Cat's not=Std, then
assign global Spec.:=jointext(jointext(global Spec.," "),Cat's).
end end
break
end
End.
```

Procedure 4: ***Print_Issued_Manuf.***

Functions: This processing procedure is for printing Manuf. Plan reports from issued Manuf. Plan file as an official report.

```
DQL QUERY
define global "Spec."Text 50.  define global "MP_No."Text 5.
define global "Issue"Text 1.  define global "QA1"Text 18.
define global "QA2"Text 18.  define global "QA3"Text 18.
define global "QA4"Text 18.  define global "Unit1"Text 18.
define global "Unit2"Text 18.  define global "Unit3"Text 18.
define global "Unit4"Text 18.  define global "Unit5"Text 18.
define global "Unit6"Text 18.  define global "Unit7"Text 18.
define global "Unit8"Text 18.  define global "Unit9"Text 18.
define global "Test1"Text 20.  define global "Test2"Text 20.
define global "Test3"Text 20.  define global "Test4"Text 20.
define global "Test5"Text 20.  define global "Test6"Text 20.
define global "Test7"Text 20.  define global "Test8"Text 20.
define global "Test9"Text 20.  define temp "WON1"Text 14.
define temp "WON2"Text 14.  define temp "WON3"Text 14.
define temp "Otherspec"text 73.  define temp "T_Draw"text 15.
define temp "FELDraw1"text 24.  define temp "FELDraw2"text 24.
define temp "FELDraw3"text 24.  define temp "C/Draw1"text 31.
define temp "C/Draw2"text 31.  define temp "C/Draw3"text 31.
define temp "Modify"text 10.  define temp "Sales1"text 8.
define temp "Sales2"text 8.  define temp "Sales3"text 8.
define temp "N"Number.
for MP_tatus with MP No. = global MP_No. and Issue = global Issue;
assign temp N:=temp N+1.
for identification with(FEL Order No.=MP_Status FEL Order No. and Ver.=MP_Status Ver.;)
for Spec._Idext with(FES=Identification FES and Group=Identification Group)
modify records
alive:=Yes.
end
case(temp N)
 value(1):
if Otherspec not=blank then
assign temp Otherspec:=jointext("Other Spec(s): ",Otherspec).
end
if length(MP_Status FEL Order No.)>=8 then
assign temp WON1:=jointext(jointext(MP_Status FEL Order No.," "),RP).
else
assign temp WON1:=jointext(jointext(firstc(MP_Status FEL Order No.,7)," "),RP).
end
assign temp Sales1:=Enquiry No..
assign temp C/Draw1:=Customer Drawing No..
assign temp FELDraw1:=Drawing No..
value(2):
if length(MP_Status FEL Order No.)>=8 then
assign temp WON2:=jointext(jointext(MP_Status FEL Order No.," "),RP).
else
assign temp WON2:=jointext(jointext(firstc(MP_Status FEL Order No.,7)," "),RP).
end
assign temp Sales2:=Enquiry No..  assign temp C/Draw2:=Customer Drawing No..
assign temp FELDraw2=Drawing No..
value(3):
```

```
if length(MP_Status FEL Order No.)>=8 then
assign temp WON3=jointext(jointext(MP_Status FEL Order No.," "),RP).
else
assign temp WON3:=jointext(jointext(firstc(MP_Status FEL Order No.,7)," "),RP).
end
assign temp Sales3:=Enquiry No..  assign temp C/Draw3=Customer Drawing No..
assign temp FELDraw3Drawing No..
end  end  end
if jointext(jointext(temp FELdraw1,temp FELDraw2,temp FELDraw3) not=blank then
assign temp T_Draw:="FEL Drawing No.".
end
for Manuf. Plan with(MP No.=global MP_No. and Issue=global Issue);
if Modify=Yes then
assign temp Modify:="(Modified)".
end
list records
MP No.;  Issue; Date; global Spec.;
temp Otherspec; temp WON1; temp WON2; temp WON3;
temp T_Draw; temp FELDraw1; temp FELDraw2; temp FELDraw3;
temp C/Draw1; temp C/Draw2; temp C/Draw3; temp Sales1;
temp Sales2; temp Sales3; temp Modify; RMC;
any MP_Status any Identification Description;
any MP_Status any Identification Inspection;
any MP_Status any Identification any Customer Customer;
any RMC-Scrap Solid; any RMC-Scrap Turnings;
any MP_Status any Identification any Chem._Range CMin;
any MP_Status any Identification any Chem._Range SiMin;
any MP_Status any Identification any Chem._Range MnMin;
any MP_Status any Identification any Chem._Range PMin;
any MP_Status any Identification any Chem._Range SMin;
any MP_Status any Identification any Chem._Range CrMin;
any MP_Status any Identification any Chem._Range MoMin;
any MP_Status any Identification any Chem._Range NiMin;
any MP_Status any Identification any Chem._Range VMin;
any MP_Status any Identification any Chem._Range AlMin;
any MP_Status any Identification any Chem._Range SbMin;
any MP_Status any Identification any Chem._Range SnMin;
any MP_Status any Identification any Chem._Range CuMin;
any MP_Status any Identification any Chem._Range CMax;
any MP_Status any Identification any Chem._Range SiMax;
any MP_Status any Identification any Chem._Range MnMax;
any MP_Status any Identification any Chem._Range PMax;
any MP_Status any Identification any Chem._Range SMax;
any MP_Status any Identification any Chem._Range CrMax;
any MP_Status any Identification any Chem._Range MoMax;
any MP_Status any Identification any Chem._Range NiMax;
any MP_Status any Identification any Chem._Range VMax;
any MP_Status any Identification any Chem._Range AlMax;
any MP_Status any Identification any Chem._Range SbMax;
any MP_Status any Identification any Chem._Range SnMax;
any MP_Status any Identification any Chem._Range CuMax;
any MP_Status any Identification any Chem._Range Ded1;
any MP_Status any Identification any Chem._Range Ded2;
any MP_Status any Identification any Chem._Range Ded3;
any MP_Status any Identification any Chem._Range Ded4;
any MP_Status any Identification any Chem._Range NMin;
```

any MP_Status any Identification any Chem._Range AsMin;
any MP_Status any Identification any Chem._Range TiMin;
any MP_Status any Identification any Chem._Range NbMin;
any MP_Status any Identification any Chem._Range Ded1Min;
any MP_Status any Identification any Chem._Range Ded2Min;
any MP_Status any Identification any Chem._Range Ded3Min;
any MP_Status any Identification any Chem._Range Ded4Min;
any MP_Status any Identification any Chem._Range H2Min;
any MP_Status any Identification any Chem._Range AsMax;
any MP_Status any Identification any Chem._Range TiMax;
any MP_Status any Identification any Chem._Range NbMax;
any MP_Status any Identification any Chem._Range Ded1Max;
any MP_Status any Identification any Chem._Range Ded2Max;
any MP_Status any Identification any Chem._Range Ded3Max;
any MP_Status any Identification any Chem._Range Ded4Max;
any MP_Status any Identification any Chem._Range H2Max;
Long:Steelmaking; Long:Steelmaking2; Block;
all Op._File with(MP No.=Manuf. Plan MP No. and Issue=Manuf. Plan Issue)Seq. in order;
all Op._File No. in group; all Op._File Operation;
all Op._File Comment; all Op._File act;
all Op._File Prod Time;
global QA1;global QA2;global QA3;global QA4;
global Unit1;global Unit2;global Unit3;global Unit4;global Unit5;
global Unit6;global Unit7;global Unit8;global Unit9;
global Test1;global Test2;global Text3;global Text4;global Text5;
global Test6;global Test7;global Text8;global Text9; long:position;
Sketch; long:position2; long:position3; Long:Comments;
Long:Comment2; Long:Comments3; Long:Comment4; Long:Comments5;
Long:Comment6; Long:Comments7; Long:Comment8; Long:Comments9.
End.

Procedure 5: *Print_MP_Format 1.*
(This procedure is similar to procedure 4 in this section for operations and comments between 13 to 35 lines.)

Procedure 6: *Print_MP_Format 2.*
(This procedure is similar to procedure 4 in this section for operations and comments less than or equal to 13 lines.)

Procedure 7: *Print_MP_Format 3.*
(This procedure is similar to procedure 4 in this section for operations and comments larger than 35 lines.)

Procedure 8: *Print_MP_Comments.*
(This procedure is similar to Procedure 4 in this section, but this procedure only prints operations and comments on second or third page.)

4-1-2-7 Repeat Proforma Manuf. Plan

Procedure 1: *Remanufacturing*

Functions: This is a command procedure which is for generating a Manuf. Plan from previous Manuf. Plans, either Issued or Standard. It runs several procedures and finally lets you to enter data in Manuf. Plan file.

```
DQL QUERY
define global "F_No."Numeric String 7. define global "L_No."Numeric String 7.
define global "Repeat"Text 6. define global "MP_No."Text 5.
define global "New MP_No."Text 5. define global "Issue"Text 1.
define global "New Issue"Text 1. define global "Number"Number.
define global "Exit"Text 3. assign global Repeat:=firstw(data-entry Repeat,1).
assign global MP_No.:="0001". assign global New Issue:="A".
assign global F_No.:=data-entry F_No..
if data-entry L_No.:=blank then
assign global L_No.:=data-entry F_No..
else
assign global L_No.:=data-entry L_No..
end
if data-entry Are you sure?=no then
exit
else
run procedure "MP_Selection".
 if global Exit="Yes" then
 exit
 end
 if data-entry Repeat=Issued_MP then
 run procedure "MP_Creation1".
 else
 run procedure "MP_Creation2".
 end
run procedure "Modify_Mp".
end.
```

Procedure 2: ***MP_Selection***

Functions: This procedure allows user to select a Manufacturing plan from either previous approved manufacturing plans or standard manufacturing plans.

```
DQL QUERY
define global "Repeat" Text 6. define global "MP_No." Text 5.
define global "Issue" Text 1. define global "New MP_No." Text 5.
define global "New Issue" Text 1. define global "Number" Number.
define global "Exit" Text 3.
while current status not=1 do
message"Press CTRL+F10 to select Manuf. plan.".
 if global Repeat="stand." then
 input using Stand._MP_Selection into "Temp_MP1".
 else
 input using Issued_MP_Selection into "Temp_MP2".
 end
if current status =1 then
assign global Exit:="Yes".
exit
else
 if Temp_MP1 MP_No. not=blank then
 assign global MP_No.:=Temp_MP1 MP_No.. assign global Issue:=Temp_MP1 Issue.
 break
 else
 if Temp_MP2 MP_No. not=blank then
 assign global MP_No.:=Temp_MP2 MP_No.. assign global Issue:=Temp_MP2 Issue.
 break
```

```
 else
 assign global Exit:="Yes".
 exit
 end  end  end
End.
```

Procedure 3: ***MP_Creation1***

Functions: This procedure is for creating a variant Manufacturing plan from previous issued Manuf. Plan as specified by users. It is run by a command procedure, Repeatmanuf, It also creates a MP status to link Manufacturing Plan with Forging Definition file for users.

```
DQL QUERY
define global "MP_No." Text 5.  define global "New MP_No." Text 5.
define global "Issue" Text 1.  define global "New Issue" Text 1.
define global "F_No." Numeric String 7.  define global "L_No." Numeric String 7.
define global "Number" Number.  define temp "ID_No." Text 5.
define temp "Ver. "Text 5.
if global New MP_No.=blank then
assign global New MP_No.:=highest of MP_Status MP No.+1.
assign global New MP_No.:=jointext(firstc("000",(4-length(global New MP_No.))),
                global New MP No.).
if global New MP_No.=blank then
exit
end
end
if global F_No.=blank then
--creating a new MP_Status records.
 for MP_Status with MP No.=global MP_No. and Issue=global Issue;
 enter a record in Renumber_Op.
 copy all from MP_Status.
end
for Renumber_Op. with MP No.=global MP_No. and Issue=global Issue;
assign temp Ver.:=Ver..
  for Identification with(FEL Order No.=Renumber_Op. FEL Order No.)
  assign temp Ver.:=Ver..
  end
enter a record in MP_Status
copy all from Renumber_Op.;
Issue :=globle New Issue;  Ver.:=temp Ver..
delete records
end
else
for status with FEL Order No. between global F_No. to global L_No.;
if temp ID_No. not =ID_No. then
assign temp ID_No.:=ID_No..
for Identification with (ID_No.=temp ID_No.)
 enter a record in MP_Status
 copy all from Identification;
 MP No.:=global New MP_No.;  Issue :=global New Issue.
end  end  end  end
  for Manuf. plan with MP No.=global MP_No. and Issue=global Issue;
  enter a record in New_Manuf. plan
  copy all from Manuf. plan;
  Date:=current date;  Date_crd:=current date;
```

```
  MP No.:=global New MP_No.;  Issue:=upper(global New Issue);
  Unique:=jointext(global New MP_No., global New Issue);  Initial:=current user name.
 for OP._file with(MP No.=global MP No. and Issue=global Issue)
  enter a record in Operation
  copy all from Op._File;
  MP No.:=global New MP_No.;  Issue:=global New Issue.
 end
 for Test_Unit with(MP No.=global MP_No. and Issue=global Issue)
  enter a record in New_Test_Unit
  copy all from Test_Unit;
  MP No.:=globle New MP_No.;  Issue:=global New Issue.
 end
 End.
```

Procedure 4: ***MP_Creation2***

Functions: This is a processing procedure for creating a variant Manufacturing plan from Master Manuf. Plans specified by users. It is run by a command procedure, Repeatmanuf.

```
DQL QUERY
define global "MP_No." Text 5.  define global "Issue" Text 1.
define global "New MP_No." Text 5.  define global "New Issue" Text 1.
define global "F_No." Numeric String 7.  define global "L_No." Numeric String 7.
define global "Number" Number.  define temp "ID_No." Text 5.
define temp "Ver. "Text 5.  define temp "Ver. "Text 5.
assign global New MP_No.:=highest of MP_Status MP No.+1.
assign global New MP_No.:=jointext(firstc("000",(4-length(global New MP_No.))), global New MP
No.).
 if global New MP_No.=blank  then
 exit
 end
for status with FEL Order No. between global F_No. to global L_No.;
if temp ID_No. not =ID_No. then
assign temp ID_No.:=ID_No..
assign temp Ver.:=Ver..
for Identification with (ID_No.=temp ID_No.and Ver.=temp Ver.)
 enter a record in MP_Status
 copy all from Identification;
 MP No.:=global New MP_No.;  Issue :=global New Issue.
end  end  end
for Standard_MPs with MP+_No.=global MP_No. and Issue=global Issue;
   enter a record in New_Manuf._plan
   copy all from Standard_MPs;
   Date:=current date;  Date_crd:=current date;
   MP No.:=global New MP_No.;  Issue:=upper(global New Issue);
   Initial:=current user name.
  for Standard_OPs
  enter a record in Operation
  copy all from Standard_OPs;
  MP No.:=global New MP_No.;  Issue:=global New Issue.
 end
 for Standard_Test
 enter a record in New_Test_Unit
 copy all from Standard_Test;
 MP No.  :=globle New MP_No.;  Issue:=globle New Issue.
 end
```

End.

Procedure 5: *Modify_Mp*

Functions: This procedure allows user to access and edit current Manuf. Plans. It can re-arrange operational sequences as specified by the users.

```
DQL QUERY
define temp "N" Number.  define temp "LSN" Number.
define temp "CN" Number.  define temp "OPN" Number.
define temp "TN" Number.
while current status not=1 do
assign temp N:=blank.  assign temp LSN:=blank.
assign temp CN:=blank.  assign temp TN:=blank.
message"Search using SHFT-F3, SHFT-F1 or ALT-F5.".
input using New_Manuf._Plan into "temp_Mp".
if current status=1 then
exit
else
if current status=3 then
message"Wait for DataEase processing please...".
for New_Manuf._Plan with(MP No.=Temp_Mp MP No. and Issue=Temp_Mp Issue);
modify records
copy all from Temp_Mp;
Initial:=current user name.
 for Operation
 if Seq.=blank then
 if No.=blank then
 if comment=blank then
 delete records
 end   end   end   end
 for New_Test_Unit
 if Test_No.=blank and Tests=blank then
 delete records
 end   end
 for Operation
 assign temp TN:=temp TN+1.
 enter a record in Renumber_Op.
 copy all from Operation.
 delete records
 end
 assign temp TN:=temp TN+2.
 while temp N not=temp TN do
 for Renumber_Op. with (MP No.=temp_Mp MP No.  and Issue=Temp_Mp Issue and Seq.=temp N)
 assign temp LSN:=temp LSN+1.
 if Operation not=blank then
 assign temp CN:=temp CN+1.  assign temp OPN:=temp CN.
 else
 assign temp OPN:=blank.
 end
 enter a record in Operation
 copy all from Renumber_Op.;
 SEQ.:=temp LSN;  No.:=temp OPN.
 delete records
 end
assign temp N:=temp N+1.
```

```
end end end end
exit
End.
```

4-1-2-8 View Issued Manuf. Plan

Procedure 1: *ViewMp*

Functions: This procedure provides only view function for user to view the issued manufacturing plans.

```
DQL QUERY
while current status not=1 do
message"Search using ALT-F5, SHFT-F3 or SHFT-F1.".
input using Manuf. Plan into "TempMp".
if current status=1 then
exit
end
 if current status=2 or current status =3 then
message"This is issued manufacturing plan.|
you are not allowed to modify it."window.
end
End.
```

4-1-3 Steel Order Menu

4-1-3-1 Start Steel Order

Procedure 1: *Melting Range*

Functions: This is a command procedure for creating Steel Order reports. It executes certain processing procedures depending on whether or not a melting range exists and satisfies the current specifications. And finally, it allows user to access and edit the Steel Order file.

```
DQL QUERY
define global "Melting No."Text 5. define global "FES"Numeric String 4.
define global "Group" 1. assign global FES:=data-entry FES.
assign global Group:=data-entry Group. define global "FEL Order No."Numeric String 11.
define global "F_No."Numeric String 7. define global "L_No."Numeric String 7.
define global "FEL Order 1"Numeric String 7. define global "FEL Order 2"Numeric String 7.
define global "FEL Order 3"Numeric String 7. define global "FEL Order 4"Numeric String 11.
define global "FEL Order No.1"Numeric String 11. define global "Exit"text 3.
define global "Number"Number. define global "New ST_No."Text 6.
define global "New Issue"Text 1. assign global New Issue:="A".
assign global FEL Order No.:=data-entry FEL Order No..
assign global F_No.:=data-entry F_No..
if data-entry L_No.=blank then
assign global L_No.:=data-entry F_No..
else
assign global L_No.:=data-entry L_No..
end
assign global FEL Order 1:=data-entry FEL Order 1.
assign global FEL Order 2:=data-entry FEL Order 2.
assign global FEL Order 3:=data-entry FEL Order 3.
assign global FEL Order 4:=data-entry FEL Order 4.
```

```
assign global Melting No.:=data-entry Melting.
if data-entry are you sure? =no then
exit
else
if firstw(data-entry Text,1)="No." and data-entry Accept=no then
exit
end
if data-entry Melting=blank then
assign global Melting No.:=00001.
else
run procedure "M2".
if current status=1 global Exit="Yes" then
exit
end
else
if  data-entry Accpet=Yes then
assign global Melting No.:=data-entry Melting.
else
assign global Melting No.:=blank.
run procedure "M1".
if current status=1 or global Exit="Yes" or global Melting No.=blank then
exit
end
end
end
run procedure "Basedata 1".
run procedure "Modify_St".
End.
```

Procedure 2: ***M1***

Functions: This procedure is for listing all existing FEL Melting Ranges being used for the specification specified in the steel order to be created, and allows user to select from these melting ranges. It also checks user's data entry to eliminate design errors such as out of range of required values.

```
DQL QUERY
define global "Melting No."Text 5. assign global Melting No.:=balnk.
define global "FEL Order No."Numeric String 11. define global "FES "Numeric String 4.
define global "Group" text 1. define global "New ST_No." Text 6.
define global "New Issue" text 1. define global "Exit" Text 3.
define temp "Enter"Text 1.
for Chem._Range with FES=global FES;
for Steelord with (FES=global FES and approve=yes)
for FEL Range with (Melting No.=Steelord Melting No.)
assign temp Enter:="Y".
for Matching_Melt.Range with(FEL Order No.=global FEL Order No.)
if Melt._No.=FEL Range Melting No. then
assign temp Enter:="N".
end  end
if temp Enter="Y" then
enter a record in Matching_Melt.Range
copy all from FEL Range;
Melt._No.:=FEL Range Melting No.; FES1:=Chem._Range FES;
Spec:=Chem._Range Specification; Source:=Chem._Range Source;
Issue:=Chem._Range Issue No.; group1:=Steelord Group;
```

```
FEL Order No.:=global FEL Order No.;  FES:=global FES;
Group:=global Group;
message:"Use CTR+F10 to view/select any other FEL Range designed for this spec.
end  end   end
message"CTRL+F10 to load FEL Melt.Range.".
input using Design_Melt.Range into "Temprange".
if current status=1 then
delete records in Matching_Melt.Range with(FEL Order No.=global FEL Order No.)
exit
end
if current status=2 then
if Temprange Are you sure?=no then
message"You should press ESC key. Exit now."window.
end
if Temprange FEL Order No. not=global FEL Order No. then
assign global Exit:="Yes".
message"You picked up a wrong Steel Order. Abort now."window.
delete records in Matching_Melt.Range with(FEL Order No.=global FEL Order No.)
exit
end
if Temprange Melt._No. not=blank and Temprange Modify =No then
assign global Melting No.:=Temprange Melt.No..
else
assign temp Enter:="Y".
for Matching_Melt.Range with(FEL Order No.=global FEL Order No.)
if Quality=Temprange Quality and
CMin=Temprange CMin and SiMin=Temprange SiMin and
MnMin=Temprange MnMin and PMin=Temprange PMin and
SMin=Temprange SMin and CrMin=Temprange CrMin and
MoMin=Temprange MoMin and NiMin=Temprange NiMin then
if VMin=Temprange VMin and AlMin=Temprange AlMin and
SbMin=Temprange SbMin and SnMin=Temprange SnMin and
CuMin=Temprange CuMin and NMin=Temprange NMin and
AsMin=Temprange AsMin and TiMin=Temprange TiMin and
NbMin=Temprange NbMin and Ded1Min=Temprange Ded1Min then
if Ded2Min=Temprange Ded2Min and Ded3Min=Temprange Ded3Min and
Ded4Min=Temprange Ded4Min and H2Min=Temprange H2Min and
CMax=Temprange CMax and SiMax=Temprange SiMax and
MnMax=Temprange MnMax and PMax=Temprange PMax then
if SMax=Temprange SMax and CrMax=Temprange CrMax and
MoMax=Temprange MoMax and NiMax=Temprange NiMax and
VMax=Temprange VMax and AlMax=Temprange AlMax and
SbMax=Temprange SbMax and SnMax=Temp_MMR SnMax then
if CuMax=Temprange CuMax and NMax=Temprange NMax and
AsMax=Temprange AsMax and TiMax=Temprange TiMax then
NbMax=Temprange NbMax and Ded1Max=Temprange Ded1Max then
Ded2Max=Temprange Ded2Max and Ded3Max=Temprange Ded3Max and
Ded4Max=Temprange Ded4Max and H2Max=Temprange H2Max then
if CAim=Temprange CAim and SiAim=Temprange SiAim and
MnAim=Temprange MnAim and PAim=Temprange PAim and
SAim=Temprange SAim and CrAim=Temprange CrAim and
MoAim=Temprange MoAim and NiAim=Temprange NiAim then
if VAim=Temprange VAim and AlAim=Temprange AlAim and
SbAim=Temprange SbAim and SnAim=Temprange SnAim and
CuAim=Temprange CuAim and NAim=Temprange NAim and
AsAim=Temprange AsAim and TiAim=Temprange TiAim and
```

```
NbAim=Temprange NbAim and Ded1Aim=TemprangeR Ded1Aim then
if Ded2Aim=Temprange Ded2Aim and Ded3Aim=Temprange Ded3Aim and
Ded4Aim=Temprange Ded4Aim and H2Aim=Temprange H2Aim and
Long:=Comment = Temprange Long:=Comment then
assign temp Enter:="N".
break
end end end end end end end end end
if temp Enter="N" then
assign global Exit:="Yes".
message"You haven't changed any data or Same to the one found."window.
delete records in Matching_Melt.Range with(FEL Order No.=global FEL Order No.)
exit
else
enter a record in FEL Range
copy all from Temprange;
Name:=current name.
assign global Melting No.=highest of FEL Range Melting No..
end end
delete records in Matching_Melt.Range with(FEL Order No.=global FEL Order No.)
exit
end
End.
```

Procedure 3: ***Basedata1***
(Same as procedure 2 listed in section 4-1-2-3.)

Procedure 4: ***M2***

Functions: This procedure is designed to search a FEL Melting Range that ten principal chemical elements are within the steel specification.

```
DQL QUERY
define global "FEL Order No."Numeric String 11. define global "FES"Numeric String 4.
define global "Group"Text 1. define Temp "T_FES"Numeric String 4.
define Temp "T_Group"Text 1. define global "Melting No."Text 5.
define temp "Enter"text 1. define temp "Design"text 1.
define global "New ST_No."Text 6. define global "New Issue"Text 1.
define global "Exit"text 3.
for Chem._Range with FES=global FES;
for FEL Range with(CMin>=Chem._Range CMin and SiMin>=Chem._Range SiMin and
MnMin>=Chem._Range MnMin and PMin>=Chem._Range PMin and
SMin>=Chem._Range SMin and CrMin>=Chem._Range CrMin)
if MoMin>=Chem._Range MoMin and NiMin>=Chem._Range NiMin and
VMin>=Chem._Range VMin and AlMin>=Chem._Range AlMin and
CMax<=Chem._Range CMax and SiMax<=Chem._Range SiMax and
MnMax<=Chem._Range MnMax and PMax<=Chem._Range PMax then
if SMax<=Chem._Range SMax and CrMax<=Chem._Range CrMax and
MoMax<=Chem._Range MoMax and NiMax<=Chem._Range NiMax and
VMax<=Chem._Range VMax and AlMax<=Chem._Range AlMax then
assign global Melting No.:=Melting No..
assign temp Enter:="N".
for Steelord with(Melting No.=global Melting No.)
if approve=Yes then
assign temp Enter:="Y".
assign Temp T_FES:=FES.
assign Temp T_Group:=group.
```

```
break
end end
for Matching_Melt.Range with(FEL Order No.=global FEL Order No.)
if Melt.No.=global Melting No. then
assign temp Enter:="N".
end
End
```

Procedure 5: ***Modify_St***

Functions: This procedure allows user to access and edit current Steel Order Plans except the max. and min. chemical composition of the melting range which can only be changed through a proper menu routine, see section 4-1-3-3, but the user can change another type of the melting range for the Steel Order Plan. Each time a modification is made, the user's name and the date are recorded. This procedure is similar to procedure 3 in section 4-1-2-3.

4-1-3-2 Create a New Issue of SO

Procedure 1: ***Modifysteel***

Functions: This is a command procedure that controls process for generating next issue of Steel Order reports. It can check user's data entry to prevent any error in this process and generate the next issue letter automatically, and finally, it lets user to access the new steel order and enter or modify data in the Steel Order File.

```
DQL QUERY
define global "New ST_No."Text 6. define global "ST_No."Text 6.
define global "New Issue"Text 1. define global "Issue"Text 1.
define global New ST_No.:=data-entry Steel No..
define global ST_No.:=data-entry Steel No..
define global New Issue:=data-entry N_I.
define global Issue:=data-entry Version.
define global "Exit"Text 3.
if data-entry Are you sure? = Yes then
run procedure "Creating_New_ST".
run procedure "ViewSt".
End.
```

Procedure 2: ***Creating_New_ST***

Functions: This procedure is for generating a new version of Steel Order and/or Ingot Requirement from the latest version of these plans.

```
DQL QUERY
define global "New ST_No."Text 6. define global "ST_No."Text 6.
define global "New Issue"Text 1. define global "Issue"Text 1.
define global "FEL Order 1"Numeric String 7. define global "FEL Order 2"Numeric String 7.
define global "FEL Order 3"Numeric String 7. define global "FEL Order 4"Numeric String 7.
define global "F_No."Numeric String 7. define global "L_No."Numeric String 7.
message"Waiting for Datafile processing...".
--temp variables, F_I is for First FEL Order Item to be processed, F for that the item is found,
--N_F for not found.
for Issued_Steelord with ST_No.=global ST_No. and Version=global Issue;
enter a record in Steelord
copy all from Issued Steelord;
```

```
ST_No.:=global New ST_No.;
version:=upper(global New Issue);
approve:=no; Date:=current date; Initial:=current user name.
end
enter a record in Ingot_Requirement
ST_No.:=global New ST_No.; Version:=global New Issue.
if global New ST_No. not=global ST_No. then
assign global New ST_No.:=highest of SO_Status ST_No.+1.
assign global New ST_No.:=jointext(firstc("00000", (6-length(global New ST_No.))),global New
ST_No.).
for status with FEL Order No. between F_No. to L_No. or
FEL Order No.=global FEL Order 1 or FEL Order No.=global FEL Order 2 or
FEL Order No.=global FEL Order 3 or FEL Order No.=global FEL Order 4;
enter a record in SO_Status
copy all from Status;
ST_No. :=global New ST_No.; Version:=global New Issue.
end
for Ingot with ST_No.=global ST_No. and Version=global Issue;
enter a record in Newingot
copy all from Ingots;
ST_No.:=global New ST_No.; Version:=global New Issue; Initial:=current user name.
end
for Newingot with ST_No.=global New ST_No. and Version=global New Issue;
enter a record in Ingots
copy all from Newingot.
delete records
end
else
for SO_Status with ST_No.=global ST_No. and Version=global Issue;
enter a record in Ingots
copy all from Newingot.
delete records
end
else
for SO_Status with ST_No.=global ST_No. and Version=global Issue;
enter a record in Newingots
copy all from SO_Status;
ST_No.:=global New ST_No.; version:=upper(global New Issue).
end
for NewStatus with ST_No.=global New ST_No. and Version=global Issue;
enter a record in SO_Status
copy all from NewStatus.
delete records
end
for Ingot with(ST_No.=global ST_No. and Version=global Issue);
enter a record in Newingot
copy all from Ingots;
Date:=current date; Initial:=current user name; Version:=global New Issue.
for Ingot_Status with(ST_No.=Ingot ST_No. And Version=Ingots Version and Letter=Ingots Letter)
enter a record in Newstatus
copy all from Ingot_Status;
Approve:=No; Version:=global New Issue.
end end
for NewIngot with(ST_No.=global New ST_No. and Version=global New Issue);
enter a record in Ingots
copy all from Newingot.
```

```
for Newstatus with(ST_No.=global New ST_No. and Version=global New Issue and
                    Letter=NewIngot Letter)
enter a record in Ingot_Status
copy all from Newstatus.
delete records
end
delete records end
End.
```

Procedure 3: ***ViewSt***
(Same as procedure 1 listed in section 4-1-3-7.)

4-1-3-3 Modify FEL Melt.Range in Current Steel Order

Procedure 1: ***Modify_Melt.Range***

Functions: This procedure is for user to access and modify melting ranges in current steel orders.

```
DQL QUERY
define temp "Modify"text 1.  define temp "Melting"text 6 .
define temp "Enter"text 1.  assign temp Modify:="Y".
message"CTRL_F10 to select and load Melt.Range.".
input using Modify_Melting_Range into "Temp_MMR".
if current status=1 then
exit
end
if current status=2 then
if Temp_MMR Are you sure?=no then
message"You should press ESC to exit. Exit now."window.
exit
end
if Temp_MMR Text3="Save" then
assign temp Enter:="Y".
else
for Steelord with(ST_No.>Temp_MMR ST_No.);
if Melting No.=Temp_MMR M_Melting No. then
assign temp Modify:="N".
break
end  end
if temp Modify="N" then
message"This Melt.Range can't be modified, because it has been|
quoted in other Steel Order.  Create a new one now if changes have been made."window.
assign temp Enter:="Y".
else
message"Saved all changes on current FEL Melt.Range datafile." window.
modify records in FEL Range with(Melting No.=Temp_MMR M_Melting No.)
copy all from Temp_MMR;
Date:=current date;  Name:=current user name.
exit
end  end
if temp Enter="Y" then
for FEL Range with Melting No.=Temp_MMR M_Melting No.;
if Quality=Temp_MMR Quality and
CMin=Temp_MMR CMin and SiMin=Temp_MMR SiMin and
MnMin=Temp_MMR MnMin and PMin=Temp_MMR PMin and
SMin=Temp_MMR SMin and CrMin=Temp_MMR CrMin then
```

```
if MoMin=Temp_MMR MoMin and NiMin=Temp_MMR NiMin and
VMin=Temp_MMR VMin and AlMin=Temp_MMR AlMin and
SbMin=Temp_MMR SbMin and SnMin=Temp_MMR SnMin and
CuMin=Temp_MMR CuMin and NMin=Temp_MMR NMin and
AsMin=Temp_MMR AsMin and TiMin=Temp_MMR TiMin then
if NbMin=Temp_MMR NbMin and Ded1Min=Temp_MMR Ded1Min and
Ded2Min=Temp_MMR Ded2Min and Ded3Min=Temp_MMR Ded3Min and
Ded4Min=Temp_MMR Ded4Min and H2Min=Temp_MMR H2Min and
CMax=Temp_MMR CMax and SiMax=Temp_MMR SiMax and
MnMax=Temp_MMR MnMax and PMax=Temp_MMR PMax and
SMax=Temp_MMR SMax and CrMax=Temp_MMR CrMax then
if MoMax=Temp_MMR MoMax and NiMax=Temp_MMR NiMax and
VMax=Temp_MMR VMax and AlMax=Temp_MMR AlMax and
SbMax=Temp_MMR SbMax and SnMax=Temp_MMR SnMax and
CuMax=Temp_MMR CuMax and NMax=Temp_MMR NMax and
AsMax=Temp_MMR AsMax and TiMax=Temp_MMR TiMax then
NbMax=Temp_MMR NbMax and Ded1Max=Temp_MMR Ded1Max then
if Ded2Max=Temp_MMR Ded2Max and Ded3Max=Temp_MMR Ded3Max and
Ded4Max=Temp_MMR Ded4Max and H2Max=Temp_MMR H2Max and
CAim=Temp_MMR CAim and SiAim=Temp_MMR SiAim and
MnAim=Temp_MMR MnAim and PAim=Temp_MMR PAim and
SAim=Temp_MMR SAim and CrAim=Temp_MMR CrAim and
MoAim=Temp_MMR MoAim and NiAim=Temp_MMR NiAim and
VAim=Temp_MMR VAim and AlAim=Temp_MMR AlAim then
if SbAim=Temp_MMR SbAim and SnAim=Temp_MMR SnAim and
CuAim=Temp_MMR CuAim and NAim=Temp_MMR NAim and
AsAim=Temp_MMR AsAim and TiAim=Temp_MMR TiAim and
NbAim=Temp_MMR NbAim and Ded1Aim=Temp_MMR Ded1Aim and
Ded2Aim=Temp_MMR Ded2Aim and Ded3Aim=Temp_MMR Ded3Aim and
Ded4Aim=Temp_MMR Ded4Aim and H2Aim=Temp_MMR H2Aim and
Long:Comment = Temp_MMR Long:=Comment then
assign temp Enter:="N".
end end end end end end end
if temp Enter="Y" then
enter a record in FEL Range
copy all from Temp_MMR;
Date:=current date; Name:=current user name.
assign temp Melting:=highest of FEL Range Melting No..
modify records in Steelord with(ST_No.=Temp_MMR ST_No. and Version=Temp_MMR Version)
Melting No.:=temp Melting.
else
message"You haven't changed any data."window.
end end
End.
```

4-1-3-4 Edit Current Steel Order Except Melt.Range

Procedure 1: ***Modify_St***

(See procedure 5 in section 4-1-3-1 for explanation of this procedure functions.)

4-1-3-5 Print a Steel Order Report

Procedure 1: ***Print_Steel_Order***

Functions: This is a command procedure for producing output of Steel Order reports. It can check user's data entry to see if it is correct. It also asks user to specify conditions such as, is the report for "Official report" or "For reference only"?, will the report printed with customer chemical range or without it?, can the steel order be issued in the computer or not. Depending on these conditions the procedure will run certain processing procedures.

```
DQL QUERY
define global"Aprve" text 1. define global"Title" text 14.
define global"ST_No." text 6. define global"ST_I" text 1.
if data-entry Are you sure?=Yes then
assign global ST_No.:=data-entry ST_No..
assign global ST_I:=data-entry VERSION.
if data-entry Ist=Yes and data-entry App=blank then
assign global Aprve:="Y".
run procedure "Issue_Steelord".
end
if global Aprve="N" then
exit
end
if data-entry Ist=Yes or data-entry App=Yes then
assign global Title:="Official Issue".
if data-entry C_R=Yes then
run procedure "SO_Report3".
else
run procedure "SO_Report4".
end
else
assign global Title:="For Ref. only".
if data-entry C_R=Yes then
run procedure "SO_Report1".
else
run procedure "SO_Report2".
end
end
End.
```

Procedure 2: Issue_Steelord

Functions: This procedure is for issuing steel order and its ingot requirement in the computer. Before the procedure dose this, it checks if both the steel order and ingot requirement have been approved by designers who are responsible for each of these plans. The procedure will then transfer the specified steel order and its ingot requirement data from current files, Steelord and Ingot_Requirement, into Issued data files, Issued_Steelord and Issued_Ingot_Requir., which therefore can be accessed by other users in other process planning areas or functions.

```
DQL QUERY
define global "Approve" Text 1. define global "ST_No. Text 6.
define global "ST_I" Text 1.
for Steelord with ST_No. =global ST_No. and Version=global ST_I;
 for Ingot_Requirement
  if Approve=Yes and Steelord Approve=Yes then
  assign global Approve:="Y".
  enter a record in Issued_Ingot_Requir.
  copy all from Ingot_Requirement.
  delete records
  else
```

```
 assign global Approve:="N".
 message "The Steel Order can't be issued because|
 the steel order or ingot requirement hasn't been approved.|
 Please contact with the authority it may concern." window.
 break
 end
end
if global Approve="Y" then
enter a record in Issued_Steelord
copy all from Steelord.
 for SO_Status
  enter a record in NewStatus
  copy all from SO_Status.
  modify records
  FIP:=Yes.
 end
for NewStatus with (ST_No.=global ST_No.and Version=global ST_I)
for SO_Status with (FEL Order No.=firstc(NewStatus FEL Order No., 7))
 if ST_No. not=global ST_No. or Version not=global ST_I then
 if RP=NewStatus RP then
 enter a record in H_SO_Status
 copy all from SO_Status.
 delete records
 end   end
end
delete records
end
delete records
end
End.
```

Procedure 3: ***SO_Report1***

Functions: This procedure is for printing current Steel Order report with Customer chemical range for "Official Issue".

```
DQL QUERY
define global"ST_No."text 6.  define global"ST_I"text 1.
define global"Title"text 14.  define temp "RMC"texe 7.
define temp "Modify"text 10.
for Steelord WITH ST_No.=global ST_No. and Version=global ST_I;
for SO_Status
for Identification with(ID_No.=SO_Status ID_No. and Ver.=SO_Status Ver.)
for MP_Status with(FEL Order No.=Identification FEL Order No. and Ver.=Identification Ver.)
for New_Manuf._Plan
assign temp RMC:=RMC.
if Modify=Yes then
assign temp Modify:="(Modified)".
end
end
for Manuf. Plan
assign temp RMC:=RMC.
if Modify =Yes then
assign temp Modify:="(Modified)".
end  end  end  end
break
```

end
list records
global; Title; P_No.; ST_No.; Version;
To Producer; Initial; Melting No.; Date;
any SO_Status any Identification any Customer Customer;
any SO_Status any Identification Application;
any SO_Status any Identification any Chem._Range Specification;
any SO_Status any Identification any Chem._Range Issue No.;
any SO_Status any Identification any Chem._Range Grade;
any SO_Status any Identification Description;
temp RMC; temp Modify; Hot To Forge; any FEL Range Melting No.; any FEL Range Quality;
any FEL Range Ded1; any FEL Range Ded2; any FEL Range Ded3; any FEL Range Ded4;
any FEL Range CMin; any FEL Range SiMin; any FEL Range MnMin; any FEL Range PMin;
any FEL Range SMin; any FEL Range CrMin; any FEL Range MoMin; any FEL Range NiMin;
any FEL Range VMin; any FEL Range AlMin; any FEL Range SbMin; any FEL Range SnMin;
any FEL Range CuMin; any FEL Range CMax; any FEL Range SiMax; any FEL Range MnMax;
any FEL Range PMax; any FEL Range SMax; any FEL Range CrMax; any FEL Range MoMax;
any FEL Range NiMax; any FEL Range VMax; any FEL Range AlMax; any FEL Range SbMax;
any FEL Range SnMax; any FEL Range CuMax; any FEL Range CAim; any FEL Range SiAim;
any FEL Range MnAim; any FEL Range PAim; any FEL Range SAim; any FEL Range CrAim;
any FEL Range MoAim; any FEL Range NiAim; any FEL Range VAim; any FEL Range AlAim;
any FEL Range SbAim; any FEL Range SnAim; any FEL Range CuAim; any FEL Range NMin;
any FEL Range AsMin; any FEL Range TiMin; any FEL Range NbMin; any FEL Range Ded1Min;
any FEL Range Ded2Min; any FEL Range Ded3Min; any FEL Range Ded4Min;
any FEL Range H2Min; any FEL Range NMax; any FEL Range AsMax; any FEL Range TiMax;
any FEL Range NbMax; any FEL Range Ded1Max; any FEL Range Ded2Max;
any FEL Range Ded3Max; any FEL Range Ded4Max; any FEL Range H2Max;
any FEL Range NAim; any FEL Range AsAim; any FEL Range TiAim; any FEL Range NbAim;
any FEL Range Ded1Aim; any FEL Range Ded2Aim; any FEL Range Ded3Aim;
any FEL Range Ded4Aim; any FEL Range H2Aim;
any FEL Range Long:Comment; Long:FEL Note; Long:FEL Note2;
Long:FEL Note3; Chemistry; Vacuum Degassed; Melt Record;
Deoxidation; Teeming; Inc.Analysis; Ultrasonic;
any SO_Status any Identification any Chem._Rnage Ded1;
any SO_Status any Identification any Chem._Rnage Ded2;
any SO_Status any Identification any Chem._Rnage Ded3;
any SO_Status any Identification any Chem._Rnage Ded4;
any SO_Status any Identification any Chem._Rnage CMin;
any SO_Status any Identification any Chem._Rnage SiMin;
any SO_Status any Identification any Chem._Rnage MnMin;
any SO_Status any Identification any Chem._Rnage PMin;
any SO_Status any Identification any Chem._Rnage SMin;
any SO_Status any Identification any Chem._Rnage CrMin;
any SO_Status any Identification any Chem._Rnage MoMin;
any SO_Status any Identification any Chem._Rnage NiMin;
any SO_Status any Identification any Chem._Rnage VMin;
any SO_Status any Identification any Chem._Rnage AlMin;
any SO_Status any Identification any Chem._Rnage SbMin;
any SO_Status any Identification any Chem._Rnage SnMin;
any SO_Status any Identification any Chem._Rnage CuMin;
any SO_Status any Identification any Chem._Rnage CMax;
any SO_Status any Identification any Chem._Rnage SiMax;
any SO_Status any Identification any Chem._Rnage MnMax;
any SO_Status any Identification any Chem._Rnage PMax;
any SO_Status any Identification any Chem._Rnage SMax;
any SO_Status any Identification any Chem._Rnage CrMax;

any SO_Status any Identification any Chem._Rnage MoMax;
any SO_Status any Identification any Chem._Rnage NiMax;
any SO_Status any Identification any Chem._Rnage VMax;
any SO_Status any Identification any Chem._Rnage AlMax;
any SO_Status any Identification any Chem._Rnage SbMax;
any SO_Status any Identification any Chem._Rnage SnMax;
any SO_Status any Identification any Chem._Rnage CuMax;
any SO_Status any Identification any Chem._Rnage NMin;
any SO_Status any Identification any Chem._Rnage AsMin;
any SO_Status any Identification any Chem._Rnage TiMin;
any SO_Status any Identification any Chem._Rnage NbMin;
any SO_Status any Identification any Chem._Rnage Ded1Min;
any SO_Status any Identification any Chem._Rnage Ded2Min;
any SO_Status any Identification any Chem._Rnage Ded3Min;
any SO_Status any Identification any Chem._Rnage Ded4Min;
any SO_Status any Identification any Chem._Rnage H2Min;
any SO_Status any Identification any Chem._Rnage NMax;
any SO_Status any Identification any Chem._Rnage AsMax;
any SO_Status any Identification any Chem._Rnage TiMax;
any SO_Status any Identification any Chem._Rnage NbMax;
any SO_Status any Identification any Chem._Rnage Ded1Max;
any SO_Status any Identification any Chem._Rnage Ded2Max;
any SO_Status any Identification any Chem._Rnage Ded3Max;
any SO_Status any Identification any Chem._Rnage Ded4Max;
any SO_Status any Identification any Chem._Rnage H2Max;
all Ingots with(ST_No.=Steelord ST_No. and Version=Steelord Version)Letter in groups
all Ingots Ingot Size in groups;
all Ingots Len/ChiWt in groups;
all Ingots L/C.W in groups;
all Ingots Quality in groups;
all Ingots Weight in groups;
all Ingots Delivery To in groups;
all Ingots TCD in groups;
all Ingots Cost in groups;
all Ingots Initial in groups;
all Ingots all Ingot_Status with(ST_No.=global ST_No. and Version=Ingot Version and
Letter=Ingot Letter) FEL Order No.;
all Ingots all Ingot_Status all SO_Status with(FEL Order No.=
firstc(Ingot_Status FEL Order No., 7)), RP).
End.

Procedure 4: ***SO_Report 2***
(Similar to procedure 3, SO_Report 1, without customer chemical range, for official issue.)

Procedure 5: ***SO_Report 3***
(Similar to procedure 3, SO_Report 1, with customer chemical range, for reference only.)

Procedure 6: ***SO_Report 3***
(Similar to procedure 3, SO_Report 1, without customer chemical range, for reference only.)

4-1-3-6 Repeat a Previous Steel Order

Procedure 1: ***Repeatsteel***

Functions: This is a command procedure for creating a new Steel Order which can be repeated from previous approved Steel Orders.

DQL QUERY
```
define global "F_No."Numeric String 7. define global "L_No."Numeric String 7.
define global "FEL Order 1"Numeric String 7. define global "FEL Order 2"Numeric String 7.
define global "FEL Order 3"Numeric String 7. define global "FEL Order 4 "Numeric String 7.
define global "New ST_No."Text 6. define global "New Issue"Text 1.
define global "ST_No."Text 6. define global "Issue"Text 1.
assign global New Issue:=A". assign global ST_No.:=data-entry Repeat_Steel_No.
assign global Issue:=data-entry Issue. assign global F_No.:=data-entry F_No..
if data-entry L_No.:=blank then
assign global L_No.:=data-entry F_No..
else
assign global L_No.:=data-entry L_No..
end
assign global FEL Order 1:=data-entry FEL Order 1.
assign global FEL Order 2:=data-entry FEL Order 2.
assign global FEL Order 3:=data-entry FEL Order 3.
assign global FEL Order 4:=data-entry FEL Order 4.
if data-entry Are you sure?=Yes then
run procedure "Creating_New_ST".
run procedure "Modify_ST".
End.
```

Procedure 2: ***Creating_New_ST***
(Same as procedure 2 listed in section 4-1-3-2.)

Procedure 3: ***Modify_ST***
(See procedure 5 in section 4-1-3-1 for explanation of functions.)

4-1-3-7 View Issued Steel Order

Procedure 1: ***Viewst***

Functions: This procedure provides view function for user to view the steel order datafiles.

DQL QUERY
```
message"Search using ALT-F5, SHFT-F3 or SHFT-F1.".
while current status not = 1 do
input using Issued_Steelord into "TempSteel".
if current status=1 then
exit
else
message"This Steel Order has been issued.|
You are not allowed to modify it."window.
exit
end
End.
```

4-1-4 Forging Procedure

4-1-4-1 Create a Forging Procedure

Procedure 1: *FP_Data_Entry*

Functions: This procedure is for creating a new record of Forging Procedures with key fields and some basic data such as, user's name, report date and etc.

```
DQL QUERY
if data-entry Are you sure?=Yes then
enter a record in FP_Datafile
Date:=current date; Name:=current name;
FP:=data-entry FP_No.; FP_I:=data-entry Issue;
Letter:=data-entry Letter; MP_I:=data-entry Issue.
for MP_Status with MP No.=data-entry FP_No. and Issue=data-entry Issue;
end
message"Record created, please go to the QP Edition Menu."window.
else
message"Not being confirmed, You should press ESC, now."window.
End.
```

4-1-4-2 Edit None Issued FP.

Procedure 1: *Edit_FP_Datafile*

Functions: This procedure is for user to access and edit a record in Forging Procedure Datafile, FP. It can also re-arrange the sequence of the FP comments as specified by the user.

```
DQL QUERY
define temp "N" Number. define temp LSN" Number.
define temp TN" Number.
while current status not=1 do
message"Search using ALT-F5, SHFT-F3 or SHFT-F1."
assign temp N:=blank. assign temp LSN:=blank.
assign temp TN:=blank.
input using FP into "Temp_FP".
if current status = 1 then
exit
end
if current status=2 or current status=3 then
message"Wait for DataEase processing please...".
for FP with(FP_No.=Temp_FP FP_No. and FP_I=Temp_FP Issue and Letter:=Temp_FP Letter);
modify records
copy all from Temp_FP;
Date:=current date; Name:=current user name.
for FP_Comments with(FP_No.=Temp_FP FP_No. and FP_I=Temp_FP Issue and
                        Letter=Temp_FP Letter)
 if Seq.=blank then
 if Comment=blank then
 delete records
 end end
end
if current status=2 then
for FP_Comments with(FP_No.=Temp_FP FP_No. and FP_I=Temp_FP Issue and Letter=Temp Letter)
 assign temp TN:=temp TN+1.
```

```
 enter a record in RN_FP
 copy all from FP_Comments.
 delete records
 end
 assign temp TN:=temp TN+2.
 while temp N not=temp TN do
for RN_FP with(Seq.=temp N and FP_No.=Temp_FP FP_No. and FP_I =Temp_FP Issue and
                              Letter=Temp Letter)
 assign temp LSN:=temp LSN+1.
 enter a record in FP_Comments
 copy all from RN_FP;
 Seq.:=temp LSN.
 delete records
end
assign temp N:=temp N+1.
end end end end
End.
```

4-1-4-3 Create a New Issue of FP.

Procedure 1: *New_Issue_FP*

Functions: This procedure is for creating a new issue of Forging Procedures based on previous issued procedures. The Forging Procedure status records will be also created for the new issue of the procedure.

```
DQL QUERY
define global "FP_No."text 5. define global "Letter"text 1. define global "Issue"text 1.
define global "New_Issue"text 1. define temp "T_I"text 1.
if data-entry Are you sure?=Yes then
assign global Issue:=data-entry Issue. assign global New_Issue:=data-entry N_I.
assign global FP_No.:=data-entry FP.
for Issued_FP with FP=global FP_No. and Letter=global Letter and Issue=global Issue;
enter a record in FP
copy all from Issued_FP;
Date:=current date; Name:=current user name; FP_I:=global New_Issue.
for Issued_FP_Comments
enter a record in FP_Table
copy all from Issued_FP_Table;
FP_I:=global New_Issue.
end
end end
message"New Issue created, please go to the Edition Menu."window.
else
message"Not being confirmed, you should press ESC, exit now."window.
End.
```

4-1-4-4 Print a Forging Procedure

Procedure: ***Print_FP***

Functions: This procedure is used to print Forging Procedure reports. It is similar to the procedure, Print_QP. listed in section 4-1-8-6.

4-1-4-5 Repeat Previous FP.

Procedure: *Repeat_FP*

Functions: This procedure is used to generate a new issue of a Forging Procedure from its previous issues. It is similar to the procedure, Repeat_Qp, listed in section 4-1-9-7.

4-1-5 Quality Plan Operation

4-1-5-1 Create a CPP Procedure
4-1-5-2 Create a Forging Procedure
4-1-5-3 Create a HT. Procedure
4-1-5-4 Create a MT. Procedure
4-1-5-5 Create a Special Procedure

(These Procedures are similar to procedure 1, FP_Data_Entry, listed in section 4-1-4-1.)

4-1-5-6 Create a QP._Operation

Procedure 1: *QP._Data_Entry*

Functions: This is a command procedure for controlling the data entry into quality plan data files. It can check user's data entry to identify whether or not the record has already existed in the database. If it is not, it will ask user if he or she wants to create the record in other files relevant to the quality plan, such as Mechanical Testing procedure or Clean/Packing/Protection procedure. It will then call certain processing procedures to create the record and allow the user to access Quality Plan to enter other information.

```
DQL QUERY
define global "OP_I"Text 1. define global "CDSL_I"Text 1. define global "MT_I"Text 1.
define global "HTP_I"Text 1. define global "SP_I"Text 1. define global "FP_I"Text 1.
define global "CPP_I"Text 1. define global "MP_No."Text 5. define global "QP_No."Text 5.
define global "FP_No."Text 5. define global "Issue"Text 1. define global "Steps"number.
define global "Batch"number.
if data-entry Are you sure? = Yes then
if data-entry OP=yes then
assign global OP_I:=data-entry Issue.
end
if data-entry CDSL=yes then
assign global CDSL_I:=data-entry Issue.
end
if data-entry MT=yes then
assign global MT_I:=data-entry Issue.
end
if data-entry HTP=yes then
assign global HTP_I:=data-entry Issue. assign global Steps:=data-entry STEPS.
end
if data-entry SP=yes then
assign global SP_I:=data-entry Issue.
end
if data-entry FP=yes then
assign global FP_I:=data-entry Issue. assign global FP_No:=data-entry MP_No..
assign global Batch:=data-entry Batch.
end
if data-entry CPP=yes then
assign global CPP_I:=data-entry Issue.
end
```

```
if data-entry MP_No.:=data-entry MP_No..
assign global QP-No.:=global MP_No.. assign global Issue:=data-entry Issue.
run procedure "Post_QP_or_FP_Data".
call menu "None Issued QP".
end.
```

Procedure 2: ***Post_QP_or_FP_Data***

Functions: This procedure is used to create records in certain quality plan files with the document key fields and other basic information before allowing the user to access and enter other information in these files such as QP_Datafile, MT_Spec., SP_Datafile and so on.

```
DQL QUERY
define global "OP_I"Text 1. define global "MT_I"Text 1. define global "CDSL_I"Text 1.
define global "HTP_I"Text 1. define global "SP_I"Text 1. define global "FP_I"Text 1.
define global "CPP_I"Text 1. define global "MP_No."Text 5. define global "QP_No."Text 5.
define global "FP_No."Text 5. define global "Issue"Text 1. define global "Steps"number.
define global "Batch"number. define temp "Name"Text 3. define temp "FES" Text 4.
define temp "Group"Text 1. define temp "No." Number. define temp "P_L"Text 1.
if global OP_I not=blank then
enter a record in QP_Datafile
QP_No.:=global QP_No.; Issue:=global Issue; MP_I:=global Issue;
Originator:=current user name; Date_crd:=current date; Date:=current date.
for MP_Status with MP No.=global MP_No. and Issue=global Issue;
enter a record in QP_Status
copy all from MP_Status;
QP_No.:=global QP_No.; Issue:=global OP_I.
end end
if global MT_I not=blank then
assign temp No.:=0.
for MP_Status with MP No.=global QP_No. and Issue=global Issue;
for Identification
assign temp FES:=FES. assign temp Group:=Group.
break
end end
enter a record in MT_Spec.
MT_No.:=global QP_No.; Issue:=global Issue; MP_I:=global MT_I;
Name:=current user name; Date:=current date; FES:=temp FES; Group:=temp Group.
end
if global SP_I not=blank then
enter a record in SP_Datafile
SP_No.:=global QP_No.; Issue:=global Issue; Name:=current user name;
Date:=current date; MP_I:=global SP_I.
end
if global CPP_I not=blank then
enter a record in CPP
CPP:=global QP_No.; Issue:=global Issue; Name:=current user name;
Date:=current date; MP_I:=global CPP_I.
end
if global HTP_I not=blank then
assign temp No.:=0. assign temp P_L:=blank.
while temp No. not=(global Steps) do
assign temp No.:=temp No.+1.
case(temp No.)
value(1): assign temp P_L:="A".
```

```
value(2): assign temp P_L:="B".
value(3): assign temp P_L:="C".
value(4): assign temp P_L:="D".
value(5): assign temp P_L:="E".
end
enter a record in HTP
HTP:=global QP_No.; Issue:=global Issue; MP_I:=global HTP_I;
Step:=temp P_L; Date:=current date; Name:=current user name.
for MP_Status with MP No.=global MP_No. and Issue=global HTP_I;
for Identification
enter a record in HTP_Status
HTP:=global MP_No.; Issue:=global HTP_I; Step:=temp P_L;
FEL Order No.:=Identification FEL Order No.; RP:=Identification RP.
end end end end
if global FP_I not=blank then
assign temp No.:=0.
assign temp P_L:=blank.
while temp No. not=(global Steps) do
assign temp No.:=temp No.+1.
case(temp No.)
value(1): assign temp P_L:="A".
value(2): assign temp P_L:="B".
value(3): assign temp P_L:="C".
value(4): assign temp P_L:="D".
value(5): assign temp P_L:="E".
end
enter a record in FP_Datafile
FP:=global FP_No.; Issue:=global Issue; Batch:=global P_L;
MP_I:=global FP_I; Name:=current user name. Date:=current date.
end
end
--Creating CDSL data file according to FEL rules.
if global CDSL_I not=blank then
enter a record in CDSL
CDSL:=global QP_No.; Issue:=global Issue; MP_I:=global CDSL_I;
Name:=current user name; Date:=current date.
assign temp No.:=0.
assign temp Name:=jointext(firstc(current user name,2),firstc(lastw(current user name,1),1)).
if global OP_I not=blank then
assign temp No.:=temp No.+1.
enter a record in CDSL_Table
CDSL:=global QP_No.; Issue:=global OP_I; No.:=temp No.;
Docu._Referece:=jointext(jointext(jointext(jointext(jointext("QP/",global QP_No.),"/"),
                         lastc(current Data,5)),"/"),temp Name);
Title:="FEL Quality Plan"; Revision:=jointext("Issue",global OP_I).
end
if global MT_I not=blank then
assign temp No.:=temp No.+1.
enter a record in CDSL_Table
CDSL:=global QP_No.; Issue:=global OP_I;
No.:=temp No.; Docu._Referece:=jointext("MT.",global QP_No.);
Title:="FEL Mechanical Testing Procedure"; Revision:=jointext("Issue",global MT_I).
end
if global HTP_I not=blank then
for HTP with HTP=global QP_No. and Issue=global HTP_I;
assign temp No.:=temp No.+1.
```

```
enter a record in CDSL_Table
CDSL:=global QP_No.; Issue:=global OP_I; No.:=temp No.;
Docu._Referece:=jointext(Jointext(jointext("HTP.",global QP_No.),"/"),HTP Step);
Title:="FEL Heat Treatment Procedure";
Revision:=jointext("Issue",global HTP_I).
end
end
if global SP_I not=blank then
assign temp No.:=temp No.+1.
enter a record in CDSL_Table
CDSL:=global QP_No.; Issue:=global OP_I;
No.:=temp No.; Docu._Referece:=jointext("SP.",global QP_No.);
Title:="FEL Steelmaking Procedure"; Revision:=jointext("Issue",global SP_I).
end
if global FP_I not=blank then
for FP_Datafile with FP_No.=global MP_No. and Issue=global FP_I;
assign temp No.:=temp No.+1.
enter a record in CDSL_Table
CDSL:=global QP_No.; Issue:=global OP_I; No.:=temp No.;
Docu._Referece:=jointext(Jointext(jointext("FP.",global QP_No.),"/"),FP_Datafile Batch);
Title:="FEL Forging Procedure"; Revision:=jointext("Issue",global FP_I).
end
end
if global CPP_I not=blank then
assign temp No.:=temp No.+1.
enter a record in CDSL_Table
CDSL:=global QP_No.; Issue:=global OP_I;
No.:=temp No.; Docu._Referece:=jointext("CPP.",global QP_No.);
Title:="FEL Clean, Protection and packing Procedure";
Revision:=jointext("Issue",global CPP_I).
end
if global MP_No. not=blank then
assign temp No.:=temp No.+1.
enter a record in CDSL_Table
CDSL:=global QP_No.; Issue:=global OP_I;
No.:=temp No.; Docu._Referece:=jointext("MP.",global MP_No.);
Title:="FEL Manufacturing Plan"; Revision:=jointext("Issue",global OP_I).
end
for QP_Status with QP_No.=global QP_No;
for Identification
if Customer Drawing No. not=blank then
assign temp No.:=temp No.+1.
enter a record in CDSL_Table
CDSL:=global QP_No.; Issue:=global OP_I;
No.:=temp No.; Docu._Referece:=Identification Customer Drawing No.;
Title:=jointext(Customer Drawing Number for",Identification FEL Order No.).
end
if Drawing No. not=blank then
assign temp No.:=temp No.+1.
enter a record in CDSL_Table
CDSL:=global QP_No.; Issue:=global QP_I;
No.:=temp No.; Docu._Referece:=Identification Drawing No.;
Title:=jointext("FEL Drawing Number for",Identification Customer Drawing No.).
end end break end
message"Record created, please go to edition menu."window.
End.
```

4-1-6 None Issued QP. Edition

4-1-6-1 Edit QP_CDSL Data File
4-1-6-2 Edit QP_CPP Data File
4-1-6-3 Edit QP_FP Data File
4-1-6-4 Edit QP_HTP Data File
4-1-6-5 Edit QP_MT Data File
4-1-6-6 Edit QP_Operation Data File
4-1-6-7 Edit QP_SP Data File

(These procedures are similar to procedure 1, Edit_FP_Datafile, listed in section 4-1-4-2.)

4-1-7 New Issue QP. Creation

4-1-7-1 New Issue of CDSL
4-1-7-2 New Issue of CPP
4-1-7-3 New Issue of FP
4-1-7-4 New Issue of HTP
4-1-7-5 New Issue of MT
4-1-7-6 New Issue of SP

(These procedures are similar to procedure 1, New_Issue_FP, listed in section 4-1-4-3.)

4-1-7-7 New Issue of QP

Procedure 1: *New_Issue_QP*

Functions: This procedure is used to create a new Quality Plan from previous issued Quality Plans. It can also create quality plan status records to create the link with Forging Definition.

```
DQL QUERY
define global "QP_No." text 5. define global "Issue" text 1.
define global "New_Issue" text 1. define temp "T_I" text 1.
if data-entry Are you sure?=Yes then
assign global Issue:=data-entry Issue.
assign global New_Issue:=data-entry N_I.
assign global QP_No.:=data-entry QP_No..
for Issued_QP with QP_No.=global QP_No. and Issue=global Issue;
enter a record in QP_Datafile
copy all from Issued_QP;
Date:=current date; Originator:=current user name; Issue:=global New_Issue.
for Issued_QP_OP
enter a record in QP_OP_Datafile
copy all from Issued QP_OP;
Issue:=global New_Issue.
end
--creating QP_Status...
for MP_Status with(MP No.=data-entry QP_No.)
if temp T_I not=blank and temp T_I not=Issue then
break
else
assign temp T_I:=Issue.
for Identification
enter a record in QP_Status
QP_No.:=data-entry QP_No.; Issue:=data-entry N_I;
```

```
FEL Order No.:=Identification FEL Order No.; Ver.:=Identification Ver..
end end end end
message"New Issue created, please go to the QP Edition Menu."window.
else
message"Not being confirmed, you should press ESC, exit now."window.
End.
```

4-1-8 Quality Plan Printing

4-1-8-1 Print a CDSL Procedure

Procedure 1: *Print_CDSL_Data*

Functions: This procedure is designed print the CDSL Process Plans.

```
DQL QUERY
define global "CDSL_No."Text 5.define global "Issue"text 1.
define global WON1"text 14. define global WON2"text 14.
define global WON3"text 14. define global WON4"text 14.
define temp "N"Number. define temp "TL"Number.
define temp "T_I"Number. define temp "P_N"Number.
for CDSL with CDSL=global CDSL_No. and Issue=global Issue;
if global WON1 =blank then
for MP_Status with(MP No.=CDSL CDSL)
if temp T_I not=blank and temp T_I not=Issue then
break
end
assign temp N:=temp N+1.
for Identification
case(temp N)
value(1):
assign temp T_I:=MP_Status Issue.
if length(FEL Order No.)>=9 then
assign global WON1:=jointext(jointext(FEL Order No.," "),RP).
else
assign global WON1:=jointext(jointext(firstc(FEL Order No.,7," "),RP).
end
value(2):
if length(FEL Order No.)>=9 then
assign global WON2:=jointext(jointext(FEL Order No.," "),RP).
else
assign global WON2:=jointext(jointext(firstc(FEL Order No.,7," "),RP).
end
value(3):
if length(FEL Order No.)>=9 then
assign global WON3:=jointext(jointext(FEL Order No.," "),RP).
else
assign global WON3:=jointext(jointext(firstc(FEL Order No.,7," "),RP).
end
value(4):
if length(FEL Order No.)>=9 then
assign global WON4:=jointext(jointext(FEL Order No.," "),RP).
else
assign global WON4:=jointext(jointext(firstc(FEL Order No.,7," "),RP).
end end end end end
for CDSL_Table
```

```
assign temp N :=No.. assign temp TL:=temp TL+1.
enter a record in RN_CDSL
copy all from CDSL_Table.
end
--Each page has 28 lines, total 3 pages.
while temp TL not=28 do
if temp TL=56 or temp TL=84 then
break
else
assign temp TL:=temp TL+1.
if temp TL>84 then
break
else
enter a record in RN_CDSL
CDSL:=global CDSL_No.; Issue:=global Issue; No.:=temp TL.
end end end end
for CDSL with CDSL=global CDSL_No. and Issue=global Issue;
assign temp P_N:=ceil(temp TL/28).
list records
CDSL; Issue; Name; Date;
global WON1; global WON2; global WON3; global WON4; temp P_N;
all RN_CDSL with(CDSL=global CDSL_No. and Issue=global Issue) No. in groups;
all RN_CDSL Docu._Reference;
all RN_CDSL Title;
all RN_CDSL M;
all RN_CDSL Revision.
end
for RN_CDSL with CDSL=global CDSL_No. and Issue=global Issue;
delete records
End.
```

4-1-8-2 Print a CPP Procedure
4-1-8-3 Print a Forging Procedure
4-1-8-4 Print a HT. Procedure
4-1-8-5 Print a MT. Procedure
4-1-8-6 Print a Special Procedure

(These procedures are similar to procedure 1, Print_CDSL_Data., listed in section 4-1-8-1.)

4-1-8-7 Print a Quality Plan

Procedure 1: ***Print_QP.***

Functions: This is a command procedure used to print Quality Plans. It can check user's data entry to see if the report has existed in the database. It can also ask user whether or not the other plans relevant to the quality plan need to be printed together such as Heat Treatment Procedure, CDSL Procedure, Special Procedure, and so on.

DQL QUERY

```
define global "QP_No." text 5. define global "Issue" text 1. define global "Name" text 18.
define global "OP_No." text 5. define global "SP_No." text 5. define global "CPP_No." text 5.
define global "CDSL_No." text 5. define global "WON1" Text 14. define global "WON2" Text 14.
define global "WON3" Text 14. define global "WON4" Text 14. define global "P_N" number.
define global "Exit" text 1. define global "Material" text 15.
define global "Discard" text 15. define global "Process" text.
if data-entry Are you sure?=Yes then
assign global Issue:=data-entry Issue.
```

```
if data-entry Plan=All then
assign global QP_No.:=data-entry QP.  assign global OP_No.:=data-entry QP.
assign global SP_No.:=data-entry QP.  assign global CPP_No.:=data-entry QP.
assign global CDSL_No.:=data-entry QP.
if data-entry Issue=Yes then
if data-entry Issued=blank then
run procedure "QP/FP_Issuing".
end
else
run procedure "Print_Title".
run procedure "Print_OP".
run procedure "Print_SP_Data".
run procedure "Print_CPP_Data".
run procedure "Print_CDSL_Data".
end
else
case(data-entry Plan)
value(QP):
assign global QP_No.:=data-entry QP.  assign global OP_No.:=data-entry QP.
if data-entry Issue=yes then
if data-entry Issued=blank then
run procedure "QP/FP_Issuing".
end
else
run procedure "Print_Title".
run procedure "Print_OP".
end
value(SP):  assign global SP_No.:=data-entry QP.
if data-entry Issue=yes then
if data-entry Issued=blank then
run procedure "QP/FP_Issuing".
end
else
run procedure "Print_SP_Data".
end
value(CPP):  assign global CPP_No.:=data-entry QP.
if data-entry Issue=yes then
if data-entry Issued=blank then
run procedure "QP/FP_Issuing".
end
else
run procedure "Print_CPP_Data".
end
value(CDSL):  assign global CDSL_No.:=data-entry QP.
if data-entry Issue=yes then
if data-entry Issued=blank then
run procedure "QP/FP_Issuing".
end
else
run procedure "Print_CDSL_Data".
end  end  end
End.
```

Procedure 2: ***QP/FP_Issuing***

Functions: This procedure is used to transfer either Quality Plan or Forging Procedure data into issued data files when they are approved and issued by process plan designers.

```
DQL QUERY
define global "QP_No." Text 5.  define global "OP_No." Text 5.  define global "MT_No." Text 5.
define global "SP_No." Text 5.  define global "HTP_No." Text 5.  define global "CPP_No." Text 5.
define global "CDSL_No." Text 5.  define global "FP_No." Text 5.
define global "Issue" Text 1.  define global "Steps" text 1.
define global "Batch" text 1.
for QP_Datafile with QP_No.=global OP_No. and Issue=global Issue;
 enter a record in Issued_QP
 copy all from QP_Datafile.
 for QP_OP_Datafile
 enter a record in Issued_QP_OP
 copy all from QP_OP_Datafile.
 delete records
 end
 delete records
end
  for MT_Spec. with MT_No.=global MT_No. and Issue=global Issue;
 enter a record in Issued_MT
 copy all from MT_Spec..
 for MT_Datafile
 enter a record in Issued_MT_Comment
 copy all from MT_Datafile.
 delete records
 end
 delete records
end
for SP_Datafile with SP_No.=global SP_No. and Issue=global Issue;
 enter a record in Issued_SP
 copy all from SP_Datafile.
 for SP_Comments
 enter a record in Issued_SP_Comment
 copy all from SP_Comments.
 delete records
 end
 delete records
end
for CDSL with CDSL=global CDSL_No. and Issue=global Issue;
 enter a record in Issued_CDSL
 copy all from CDSL.
 for CDSL_Table
 enter a record in Issue_CDSL_Table
 copy all from CDSL_Table.
 delete records
 end
 delete records
end
for CPP with CPP=global CPP_No. and Issue=global Issue;
 enter a record in Issued_CPP
 copy all from CPP.
 for CPP_Table
 enter a record in Issued_CPP_Table
```

```
  copy all from CPP_Table.
  delete records
  end
  delete records
 end
 for HTP with HTP=global HTP_No. and Issue=global Issue and Step=global Steps;
  enter a record in Issued_HTP
  copy all from HTP.
  for HTP_Table enter a record in Issued_HTP_Table
  copy all from HTP_Table.
  delete records
  end
  delete records
 end
 for FP_Datafile with FP_No.=global FP_No. and Issue=global Issue and Batch=global Batch;
  enter a record in Issued_FP
  copy all from FP_Datafile.
  for FP_Comment
  enter a record in Issued_FP_Table
  copy all from FP_Comment.
  end
  delete records
 End.
```

Procedure 3: *Print_Title*

Functions: This is a processing procedure used to print the front page of a Quality Plan showing some basic information such as designer's name, date of issue, and information from Forging Definition file.

```
DQL QUERY
define global "QP_No." text 5.  define global "Issue" text 1.
define global "WON1" Text 14.  define global "WON2" Text 14.
define global "WON3" Text 14.  define global "WON4" Text 14.
define global "Name" text 18.  define temp "Ingot" text 10.
define temp "Material"text 20.  define temp "Discard"text 30.
define temp "Process"text 30.  define temp "No."Number.
define global "P_N" number.
for QP_Datafile with QP_No.=global QP_No. and Issue=global Issue;
assign temp Process:=Process.  assign temp Discard:=Min_Discard.
for QP_OP_Datafile
if OP.=blank then
assign global P_N:=global P_N+2.
else
assign global P_N:=global P_N+4.
end
end
assign global P_N:=ceil(global P_N/36+1).
assign global Name:=jointext(firstc(Originator,2), firstc(lastw(Originator,1),2))
assign global Name:=jointext(jointext(jointext(jointext(jointext("QP/",QP_No.),"/"),
                    lastc(Date_Crd,5)),"/").global Name).
for QP_Status with(QP_No.=global QP_No. and Issue=global Issue)
for Identification
assign temp No.:=temp No.+1.
case(temp No.)
value(1):
```

```
if length(FEL Order No.)>=9 then
assign global WON1:=jointext(jointext(FEL Order No.," "),RP).
else
assign global WON1:=jointext(jointext(firstc(FEL Order No.,7)," "),RP).
end
value(2):
if length(FEL Order No.)>=9 then
assign global WON2:=jointext(jointext(FEL Order No.," "),RP).
else
assign global WON2:=jointext(jointext(firstc(FEL Order No.,7)," "),RP).
end
value(3):
if length(FEL Order No.)>=9 then
assign global WON3:=jointext(jointext(FEL Order No.," "),RP).
else
assign global WON3:=jointext(jointext(firstc(FEL Order No.,7)," "),RP).
end
value(4):
if length(FEL Order No.)>=9 then
assign global WON4:=jointext(jointext(FEL Order No.," "),RP).
else
assign global WON4:=jointext(jointext(firstc(FEL Order No.,7)," "),RP).
end end end end
for SO_Status with(FEL Order No.=firstc(global WON1,7))
for Steelord
for FEL range
assign temp Material:=Quality.
end
for Ingot_Requirement
for Ingots
assign temp Ingot:=Ingot Size.
break
end end end end
list records
global Name; QP_No.; Issue; Originator; Date; Min_Discard; Insp._Notice1; Insp._Notice2;
global P_N; global WON1; global WON2; global WON3; global WON4; temp Ingot;
temp Material; temp Process; temp Discard; Long:Remarks; Long:Remarks2; Long:Remarks3;
all QP_Status FEL Order No.;
all QP_Status any Identification RP;
all QP_Status any Identification any Customer Customer;
all QP_Status any Identification Customer Order No.;
all QP_Status any Identification Description;
all QP_Status any Identification Enquiry No..
End.
```

Procedure 4: ***Print_OP.***

Functions: This is a processing procedure used to print the operation part of a Quality Plan report.

```
DQL QUERY
define global "OP_No." text 5. define global "Issue" text 1. define global "Name" text 18.
define global "WON1" Text 14. define global "WON2" Text 14. define global "WON3" Text 14.
define global "WON4" Text 14. define temp "N"Number. define temp "TL"Number.
define global "P_N" number.
for QP_Datafile with QP_No.=global OP_No. and Issue=global Issue;
for QP_OP_Datafile
```

```
if Op. not=blank then
assign temp N:=Op.. assign temp TL:=temp TL+4.
else
assign temp TL:=temp TL+2.
end
enter a record in RN_QP.
copy all from QP_OP_Datafile;
Op.:=temp N;
seq.:=temp N.
end
--each page, 36 lines. Total pages are 6.
while temp TL not=36 do
if temp TL=36 or temp TL=72 or temp TL=108 or temp TL=136 or temp TL=180 or
                         temp TL=216 or temp TL=252 or temp TL=288 then
break
else
assign temp TL:=temp TL+2.
if temp TL > 288 then
break
else
enter a record in RN_QP.
QP_No.:=QP_Datafile QP_No.; Issue:=QP_Datafile Issue; Op.:=temp N; Seq.:=temp N.
end end end
for QP_Datafile with QP_No.=global OP_No. and Issue=global Issue;
list records
global Name; QP_No.; Issue; Originator; Date; global P_N;
global WON1; global WON2; global WON3; global WON4;
Inspect1; Inspect2; Inspect3; Inspect4; Inspect5;
all RN_QP. with(QP_No.=QP_Datafile QP_No. and Issue=QP_Datafile Issue) Seq. in groups;
all RN_QP. M;
all RN_QP. Op.;
all RN_QP. Operation;
all RN_QP. Docu_Ref.;
all RN_QP. Docu_Ref.;
all RN_QP. Docu_Rec.;
all RN_QP. Ins1;
all RN_QP. Ins2;
all RN_QP. Ins3;
all RN_QP. Ins4;
all RN_QP. Ins5;
end
for RN_QP. with QP_No.=global OP_No. and Issue=global Issue;
delete records
End.
```

4-1-9 Repeat Previous QP

4-1-9-1 Create a New CDSL
4-1-9-2 Repeat Previous CPP.
4-1-9-3 Repeat Previous FP.
4-1-9-4 Repeat Previous HT.
4-1-9-5 Repeat Previous MT.
4-1-9-6 Repeat Previous SP.

(The procedures above are similar to procedure 1, Repeat_Qp, listed in section 4-1-9-7.)

4-1-9-7 Repeat Previous QP_OP.

Procedure 1: ***Repeat_Qp***

Functions: This procedure is designed for generating a new Quality Plan from previous proforma of Quality Plans.

```
DQL QUERY
define global "Qp_I"Text 1. define global "MP_No."Text 5.
define global "Qp_No."Text 5. define global "Issue"Text 1.
if data-entry Are you sure?=Yes then
for Issue_Qp with(Qp_No.=data-entry p_Qp_No. and Issue=data-entry p_Issue);
enter a record in Qp_Datafile
copy all from Issued_Qp;
Qp_No.:=data-enrty MP_No.; Issue:=data-entry Issue;
MP_I:=data-entry Issue; Date:=current date; Originator:=current user name.
for Issue_Qp_Op
enter a record in Qp_Op_Datafile
copy all from Issued_Qp_Op;
Qp_No.:=data-enrty MP_No.; Issue:=data-entry Issue;
end
end
for MP_Status with(MP_No.=data-entry MP_No. and Issue=data-entry Issue);
enter a record in Qp_Status
Qp_No.:=data-entry MP_No.; Issue:=data-entry Issue;
FEL Order No.:=MP_Status FEL Order No.; Ver.:=MP_Status Ver..
end
message"Record created. please go to the Qp Edition Menu."window.
else
message"Not being confirmed, you should press [ESC], exit now."window.
End.
```

4-1-10 Ingot Requirement Menu

4-1-10-1 Design/Edit Current Ingot Requirement

Procedure 1: ***Modify_Ingot***

Functions: This procedure is designed for maintaining the Ingot Requirement data file. It allows user to load an Ingot Requirement record into a temporary file, TempIn, for user to modify. This procedure also checks if the bundle of FEL Order Numbers is valid or not for the designed ingot.

```
DQL QUERY
define temp "F" Text 1.
while current status not=1 do
assign temp F:="N".
message "Search using ALT-F5, SHFT-F3 or SHFT-F1. ".
input using Ingot_Requirement into "TempIn.".
if current status =1 then
exit
end
if current status =3 or current status =2 then
for Ingot_Requirement with ST_No.=TempIn. ST_No. and Version=TempIn. Version;
 modify records
 copy all from TempIn.
```

```
for Ingots
if Letter=blank and Ingot Size=blank and Len/Chiwt.=blank and I/C. W=blank then
delete records
else
modify records
Initial :=current user name.
end
end
for Ingot_Status
if Letter=blank and FEL Order No.=blank then
delete records
else
--check steel order status to see whether the items exist or not.
for SO_Status with (ST_No.=Ingot_Status ST_No. and Version=Ingot_Status Version
and FEL Order No.=firstc(Ingot_Status FEL Order No.,7))
 assign temp F:="Y".
end
 if temp F="N" then
message "one of the FEL Order No.(s) doesn't match with|
                  the Steel Order. Please check. "window.
end  end  end end end
End.
```

4-1-10-2 Print Steel Order Report

(Same as the menu in section 4-1-3-5, Print_Steel_Order)

4-1-10-3 View Issued Ingot Requirement Datafile
(Through a data entry menu, users in this area have lower security levels than metallurgists in Steel Order planning area.)

4-2 Procedures in Quality Assurance Functions

4-2-1 Knowledge Data Bank

4-2-1-1 Edit Un-circulated Spec.

Procedure: *Modify_Spec.*

Functions: This procedure is for user to access and modify un-circulated specifications.

```
DQL QUERY
define temp "Modify1"Text 1.  define temp "Modify2"Text 1.
define temp "Modify3"Text 1.  define temp "Modify4"Text 1.
define temp "Group"Text 1.  define global "FES"Numeric String 4.
define global "Source"Text 7.  define global "Spec"Text 15.
define global "Grade"Text 10.  define global "Issue"Text 7.
if data-entry Are you sure?=Yes then
if data-entry alive=Yes and data-entry Mod=Yes then
assign temp Group:=data-entry N_G.
enter a record in Spec._Index
copy all from data-entry;
Group:=temp Group.
else
assign temp Group:=data-entry Group.
end
```

```
if data-entry Group=temp Group and data-entry alive=no then
modify records in Chem._Range with(FES=data-entry FES)
copy all from data-entry.
modify records in Spec._Index with(FES=data-entry FES and Group=temp Group)
copy all from data-entry.
if data-entry Create_Property=no then
modify records in Spec._Index with(FES=data-entry FES and Group=temp Group)
copy all from data-entry.
T_Unit:=MPa;  Stress:=Rp0.2;  I_Unit:=J;  HB/HV:=HB;  T_Name:=C.
end
end
if data-entry Dir not=XX then
for Mech._Property with(FES=data-entry FES and Group=temp Group and No.=1);
assign temp Modify1:="Y".
modify records
copy all from data-entry.
Group:=temp Group;  No.:=data-entry Mod_No..
end
else
delete records in Mech._Property with (FES=data-entry FES and Group=temp Group and No.=1)
end
--If conditions below are true, that means the Mech._Property file doesn't have corresponding data
--with data entry form and the data in data-entry form needs to be entered into Mech._Property file.
if temp Modify1 not="Y" and data-entry Dir not=XX then
enter a record in Mech._Property
copy all from data-entry;
Group:=temp Group;  No.:=data-entry Mod_No..
end
if data-entry Dir_2 not=XX then
for Mech._Property with (FES=data-entry FES and Group=data-entry Group and No.=2;
assign temp Modify2:="Y".
modify records
copy all from data-entry.
Group:=temp Group; No.:=data-entry Mod_No.2;
Dir:=data-entry Dir_2; Locat:=data-entry Locat2;
Spec:=data-entry Spec2; TenTemp:=data-entry TenTemp2;
RpMin.:=data-entry RpMin.2; RpMax.:=data-entry RpMax.2;
RmMin.:=data-entry RmMin.2; RmMax.:=data-entry RmMax.2;
A.:=data-entry A.2; Z.:=data-entry Z.2;
ImpTemp:=data-entry ImpTemp2; Value:=data-entry Value2;
RV:=data-entry RV2; Angle:=data-entry Angle2;
Radius:=data-entry Radius2; HardMin.:=data-entry HardMin.2;
HardMax.:=data-entry HardMax.2; Shear:=data-entry Shear2;
FATT:=data-entry FATT2; HardMax.:=data-entry L.E2;
comment:=data-entry Comment2;
end
else
delete records in Mech._Property with (FES=data-entry FES and Group=temp Group and No.=2)
end
if temp Modify2 not="Y" and data-entry Dir_2 not=XX then
enter a record in Mech._Property
copy all from data-entry.
Group:=temp Group; No.:=data-entry Mod_No.2;
Dir:=data-entry Dir_2; Locat:=data-entry Locat2;
Spec:=data-entry Spec2; TenTemp:=data-entry TenTemp2;
RpMin.:=data-entry RpMin.2; RpMax.:=data-entry RpMax.2;
```

```
RmMin.:=data-entry RmMin.2; RmMax.:=data-entry RmMax.2;
A.:=data-entry A.2; Z.:=data-entry Z.2;
ImpTemp:=data-entry ImpTemp2; Value:=data-entry Value2;
RV:=data-entry RV2; Angle:=data-entry Angle2;
Radius:=data-entry Radius2; HardMin.:=data-entry HardMin.2;
HardMax.:=data-entry HardMax.2; Shear:=data-entry Shear2;
FATT:=data-entry FATT2; HardMax.:=data-entry L.E2;
Comment:=data-entry Comment2;
end
if data-entry Dir_3 not=XX then
for Mech._Property with (FES=data-entry FES and Group=data-entry Group and No.=3;
assign temp Modify3:="Y".
modify records
copy all from data-entry.
Group:=temp Group;  No.:=data-entry Mod_No.3;
Dir:=data-entry Dir_3;  Locat:=data-entry Locat3;
Spec:=data-entry Spec3;  TenTemp:=data-entry TenTemp3;
RpMin.:=data-entry RpMin.3;  RpMax.:=data-entry RpMax.3;
RmMin.:=data-entry RmMin.3;  RmMax.:=data-entry RmMax.3;
A.:=data-entry A.3;  Z.:=data-entry Z.3;
ImpTemp:=data-entry ImpTemp3;  Value:=data-entry Value3;
RV:=data-entry RV3;  Angle:=data-entry Angle3;
Radius:=data-entry Radius3;  HardMin.:=data-entry HardMin.3;
HardMax.:=data-entry HardMax.3;  Shear:=data-entry Shear3;
FATT:=data-entry FATT3;  HardMax.:=data-entry L.E3;
Comment:=data-entry Comment3;
end
else
delete records in Mech._Property with (FES=data-entry FES and Group=temp Group and No.=3)
end
if temp Modify3 not="Y" and data-entry Dir_3 not=XX then
enter a record in Mech._Property
copy all from data-entry.
Group:=temp Group; No.:=data-entry Mod_No.3;
Dir:=data-entry Dir_3; Locat:=data-entry Locat3;
Spec:=data-entry Spec3; TenTemp:=data-entry TenTemp3;
RpMin.:=data-entry RpMin.3; RpMax.:=data-entry RpMax.3;
RmMin.:=data-entry RmMin.3; RmMax.:=data-entry RmMax.3;
A.:=data-entry A.3; Z.:=data-entry Z.3;
ImpTemp:=data-entry ImpTemp3; Value:=data-entry Value3;
RV:=data-entry RV3; Angle:=data-entry Angle3;
Radius:=data-entry Radius3; HardMin.:=data-entry HardMin.3;
HardMax.:=data-entry HardMax.3; Shear:=data-entry Shear3;
FATT:=data-entry FATT3; HardMax.:=data-entry L.E3;
Comment:=data-entry Comment3;
end
if data-entry Dir_4 not=XX then
for Mech._Property with (FES=data-entry FES and Group=data-entry Group and No.=4;
assign temp Modify4:="Y".
modify records
copy all from data-entry.
Group:=temp Group; No.:=data-entry Mod_No.4;
Dir:=data-entry Dir_4; Locat:=data-entry Locat4;
Spec:=data-entry Spec4; TenTemp:=data-entry TenTemp4;
RpMin.:=data-entry RpMin.4; RpMax.:=data-entry RpMax.4;
RmMin.:=data-entry RmMin.4; RmMax.:=data-entry RmMax.4;
```

```
A.:=data-entry A.4; Z.:=data-entry Z.4;
ImpTemp:=data-entry ImpTemp4; Value:=data-entry Value4;
RV:=data-entry RV4; Angle:=data-entry Angle4;
Radius:=data-entry Radius4; HardMin.:=data-entry HardMin.4;
HardMax.:=data-entry HardMax.4; Shear:=data-entry Shear4;
FATT:=data-entry FATT4; HardMax.:=data-entry L.E4;
Comment:=data-entry Comment4;
end
else
delete records in Mech._Property with (FES=data-entry FES and Group=temp Group and No.=4)
end
if temp Modify4 not="Y" and data-entry Dir_4 not=XX then
enter a record in Mech._Property
copy all from data-entry.
Group:=temp Group; No. :=data-entry _Mod_No.4;
Dir:=data-entry Dir_4; Locat:=data-entry Locat4;
Spec:=data-entry Spec4; TenTemp:=data-entry TenTemp4;
RpMin.:=data-entry RpMin.4; RpMax.:=data-entry RpMax.4;
RmMin.:=data-entry RmMin.4; RmMax.:=data-entry RmMax.4;
A.:=data-entry A.4; Z.:=data-entry Z.4;
ImpTemp:=data-entry ImpTemp4; Value:=data-entry Value4;
RV:=data-entry RV4; Angle:=data-entry Angle4;
Radius:=data-entry Radius4; HardMin.:=data-entry HardMin.4;
HardMax.:=data-entry HardMax.4; Shear:=data-entry Shear4;
FATT:=data-entry FATT4; HardMax.:=data-entry L.E4;
Comment:=data-entry Comment4;
end
End.
```

4-2-1-2 Create a Specification

Procedure: *Spec._Creation*

Functions 1: This is a command procedure for creating a new specification. It can check user's data entry to inform user whether or not the specification has already existed in the database. It can run different processing procedures to create the specification with basic information and let user to enter other data into it, or to convert the previous specification into a new grade or issue of the specification.

```
DQL QUERY
define global "Spec"Text 15. define global "Grade"Text 10. define global "Issue"Text 7.
define global "Source"Text 7. define global "FES" Numeric String 4. define global "pregrade"Text 10.
define global "preissue"Text 7. define global FES:=data-entry FES.
define global Spec:=data-entry Spec. define global Issue:=data-entry Issue.
define global Grade:=data-entry Grade. define global Source:=data-entry Source.
define global Pregrade:=data-entry Pregrade. define global Preissue:=data-entry Preissue.
define global "Exit"Text 1.
if data-entry are you sure? not=Yes then
exit
end
if firstw(data-entry exist1,1)="March" and
  firstw(data-entry exist2,1)="March" and
  firstw(data-entry exist3,1)="March" then
run procedure "Modify_Spec.".
else
run procedure "Enter_Spec.".
```

```
if data-entry PreGrade not=blank and data-entry PreIssue
not=blank then
run procedure "Convert_Spec.".
message"Press SH-F3 once Spec._Index is loaded.".
record entry "Spec._Index".
else
run procedure "Convert_Spec.".
if current status=1 or current status=4 or global Exit="Y" then
exit
else
message"Press SH-F3 once Spec._Index is loaded.".
record entry "Spec._Index".
end
end
End.
```

Procedure 2: *Convert_Spec.*

Functions: This procedure is used to convert a previous specification into a new grade or issue of the specification.

```
DQL QUERY
define temp "Enter"Text 3. define global "FES"Numeric String 4.
define global "Source"Text 7.  define global "Spec."Text 15.
define global "Grade"Text 10.  define global "Issue"Text 7.
define global "Exit"Text 1.  define global "PreGrade"Text 10.
define global "PreIssue"Text 7.
if global PreGrade not=blank and global PreIssue not=blank then
--copy the previous record into chemical and mechanical file.
for Chem._Range with Source=global Source and Specification=global Spec and
                           Grade=global PreGrade and Issue No.=global PreIssue;
enter a record in Chem.2
copy all from Chem. Range;
FES := global FES; Grade :=global Grade;  Issue No.=global Issue; Date :=current date.
for Mech._Property with(FES=Chem._Range FES and Group="A")
enter a record in Mech.2
copy all from Mech._Property;
FES := global FES; Group:="A".
end
end
for Chem.2;
assign temp Enter :="Y".
enter a record in Chem._Range
copy all from Chem.2.
enter a record in Spec._Index
FES := global FES; Group:="A";  alive:=no; Date:=current date;
Name:=current user name.
for Mech.2
enter a record in Mech. Property
copy all from Mech.2.
delete records
end
delete records end
if temp Enter not = "Y" then
message"Wrong Previous Grade or Issue specified."window.
end
```

```
end
if temp Enter not="Y" then
enter a record in Spec._Index
FES:=global FES; Group:="A"; T_Unit:=MPa; Steress:=Rp0.2;
I_Unit:=J; HB/HV:=HB; T_Name:=C; alive:=no;
Date:=current date; name:=current user name.
enter a record in Chem._Range
FES:=global FES; Source:=global Source;
Specification:=global Spec.; Issue_No.:=global Issue;
Grade:=global Grade.
End.
```

Procedure 3: ***Enter_Spec.***

Functions: This processing procedure is for creating a new record in specification data files, Spec._Index, Chem. Range and Mech. Property with key fields and basic information entered in the data-entry form of the procedure by the user.

```
DQL QUERY
define temp "Fnd1"Text 3. define temp "Fnd2"Text 3. define temp "Fnd3"Text 3.
define temp "Fnd4"Text 3. define global "FES"Numeric String 4. define global "Source"Text 7.
define global "Spec"Text 15. define global "Grade"Text 10. define global "Issue"Text 7.
define global "Exit"Text 1.
if data-entry Are you sure?=Yes then
if data-entry Group="A" and data-entry alive not=Yes then
modify records in Chem._Range with(FES=data-entry FES)
copy all from data-entry.
modify records in Spec._Index with(FES=data-entry FES and Group=data-entry Group)
copy all from data-entry.
if data-entry Create_Property=no then
modify records in Spec._Index with(FES=data-entry FES and Group=data-entry Group)
copy all from data-entry.
T_Unit:=MPa; Steress:=Rp0.2; I_Unit:=J; HB/HV:=HB; T_Name:=C;
exit
end
if data-entry Dir not=XX then
for Mech._Property with FES=data-entry FES and Group=data-entry Group and No.=1;
assign temp Fnd1="Yes".
end
if temp Fnd1="Yes" then
modify records in Mech._Property with (FES=data-entry FES and Group=data-entry Group and No.=1)
copy all from data-entry.
else
enter a record in Mech._Property
copy all from data-entry.;
RpMin.:=data-entry RpMin.; RpMax.:=data-entry RpMax.; RmMin.:=data-entry RmMin.;
RmMax.:=data-entry RmMax.; Value:=data-entry Value.
end
end
if data-entry Dir_2 not=XX then
for Mech._Property with FES=data-entry FES and Group=data-entry Group and No.=2;
assign temp Fnd2="Yes".
end
if temp Fnd2="Yes" then
modify records in Mech._Property with (FES=data-entry FES and Group=data-entry Group and No.=2)
copy all from data-entry.
```

```
No. :=data-entry No.2; Dir:=data-entry Dir_2; Locat:=data-entry Locat2;
Spec:=data-entry Spec2; TenTemp:=data-entry TenTemp2; RpMin.:=data-entry RpMin.2;
RpMax.:=data-entry RpMax.2; RmMin.:=data-entry RmMin.2; RmMax.:=data-entry RmMax.2;
A.:=data-entry A.2; Z.:=data-entry Z.2; ImpTemp:=data-entry ImpTemp2;
Value:=data-entry Value2; RV:=data-entry RV2; Angle:=data-entry Angle2;
Radius:=data-entry Radius2; HardMin.:=data-entry HardMin.2; HardMax.:=data-entry HardMax.2;
Shear:=data-entry Shear2; FATT:=data-entry FATT2; HardMax.:=data-entry L.E2;
Comment:=data-entry Comment2;
else
end end
if data-entry Dir_3 not=XX then
for Mech._Property with FES=data-entry FES and Group=data-entry Group and No.=3;
assign temp Fnd2="Yes".
end
if temp Fnd3="Yes" then
modify records in Mech._Property with (FES=data-entry FES and Group=data-entry Group and No.=3)
copy all from data-entry.
No. :=data-entry No.3; Dir:=data-entry Dir_3; Locat:=data-entry Locat3;
Spec:=data-entry Spec3; TenTemp:=data-entry TenTemp3; RpMin.:=data-entry RpMin.3;
RpMax.:=data-entry RpMax.3; RmMin.:=data-entry RmMin.3; RmMax.:=data-entry RmMax.3;
A.:=data-entry A.3; Z.:=data-entry Z.3; ImpTemp:=data-entry ImpTemp3;
Value:=data-entry Value3; RV:=data-entry RV3; Angle:=data-entry Angle3;
Radius:=data-entry Radius3; HardMin.:=data-entry HardMin.3; HardMax.:=data-entry HardMax.3;
Shear:=data-entry Shear3; FATT:=data-entry FATT3; HardMax.:=data-entry L.E3;
Comment:=data-entry Comment3;
else
end end
if data-entry Dir_4 not=XX then
for Mech._Property with FES=data-entry FES and Group=data-entry Group and No.=4;
assign temp Fnd2="Yes".
end
if temp Fnd4="Yes" then
modify records in Mech._Property with (FES=data-entry FES and Group=data-entry Group and No.=4)
copy all from data-entry.
No. :=data-entry No.4; Dir:=data-entry Dir_4; Locat:=data-entry Locat4;
Spec:=data-entry Spec4; TenTemp:=data-entry TenTemp4; RpMin.:=data-entry RpMin.4;
RpMax.:=data-entry RpMax.4; RmMin.:=data-entry RmMin.4; RmMax.:=data-entry RmMax.4;
A.:=data-entry A.4; Z.:=data-entry Z.4; ImpTemp:=data-entry ImpTemp4;
Value:=data-entry Value4; RV:=data-entry RV4; Angle:=data-entry Angle4;
Radius:=data-entry Radius4; HardMin.:=data-entry HardMin.4; HardMax.:=data-entry HardMax.4;
Shear:=data-entry Shear4; FATT:=data-entry FATT4; HardMax.:=data-entry L.E4;
Comment:=data-entry Comment4;
else
enter a record in Mech._Property
copy all from data-entry.
No. :=data-entry No.4; Dir:=data-entry Dir_4; Locat:=data-entry Locat4; Spec:=data-entry Spec4;
TenTemp:=data-entry TenTemp4; RpMin.:=data-entry RpMin.4; RpMax.:=data-entry RpMax.4;
RmMin.:=data-entry RmMin.4; RmMax.:=data-entry RmMax.4; A.:=data-entry A.4;
Z.:=data-entry Z.4; ImpTemp:=data-entry ImpTemp4; Value:=data-entry Value4;
RV:=data-entry RV4; Angle:=data-entry Angle4; Radius:=data-entry Radius4;
HardMin.:=data-entry HardMin.4; HardMax.:=data-entry HardMax.4; Shear:=data-entry Shear4;
FATT:=data-entry FATT4; HardMax.:=data-entry L.E4; Comment:=data-entry Comment4;
end end
else
assign global Exit:="Yes".
End.
```

4-2-1-3 Create a New Version of Spec

Procedure 1: ***Chemitospec***

Functions: This is a command procedure designed to create a new version of a specification based on previous specification.

DQL QUERY

```
define global "Unit1"Text 12.  define global "Unit2"Text 12.  define global "Spec"Text 15.
define global "Grade"Text 10.  define global "Issue"Text 7.  define global "Source"Text 7.
define global "FES" Numeric String 4.  define global "Pregrade"Text 10.
define global "Preissue"Text 7.  Assign global Unit1:=data-entry Unit1.
assign global Unit2:=data-entry Unit2.  assign global FES:=data-entry FES.
assign global Spec:=data-entry Spec.  assign global Issue:=data-entry Issue.
assign global Grade:=data-entry Grade.  assign global Source:=data-entry Source.
assign global Pregrade:=data-entry Pregrade.  assign global Preissue:=data-entry Preissue.
define global "Exit"Text 1.
if data-entry sure not=yes then
exit
else
if firstw(data-entry Ready,2) not="Press F2" then exit else
if firstw(data-entry message1, 1)=firstw(data-entry message2, 1) then
run procedure "Specmodiapprove".
else
if firstw(data-entry message2, 1)="choose" then
if data-entry Unit1=blank or
  data-entry Unit2=blank then
message"The Tensile or Impact Unit is not entered."window.
exit
else
run procedure "Enterspec".
if current status=1 or current status=4 or global Exit="Y" then
exit
end  end
else
run procedure "Modispec".
run procedure "Specmodiapprove".
end  end  end
End.
```

Procedure 2: ***Specmodiapprove***

Functions: This is a processing procedure for controlling the process of modifications to the Mechanical Properties.

DQL QUERY

```
define temp "Modify"text 1.  define temp "Enter"text 1.
while current status not=1 do
assign temp Modify:="Y".
message"Press SH-F3 once Mechproperty is loaded.".
input using Mechproperty into "TempSpec".
if current status=1 then
exit
end
```

```
if current status=2 then
enter a record in Mechproperty
copy all from TempSpec;
alive:= No;  Name:=current user name;  Date:=current date.
break
end
for Mechproperty with FES=TempSpec FES and Group=TempSpec Group;
--if alive file in Mechproperty file is "Yes", it means the Spec. is in use.
if alive=yes then
message"This specification has already been circulated and is in use.|
You are not allowed to modify it or delete it.|
It can only be withdrawn and/or updated to a new issue |
by the QA authority." window.
assign temp Modify:="N".
else
--if alive is "No" and the spec. has been used, it means the spec. is revised.
for Identification
if Status=yes then
assign temp Modify:="N".
message"This specification issue has been withdrawn.|
You are not allowed to modify it.|
It can be put back in use and/or updated to a new issue|
but only by the QA authority."window.
delete records in S_Spec2 with(FES+TempSpec FES)
break
end  end  end
if current user level<="3" and alive not=TempSpec alive then
modify records in Mechproperty with(FES=TempSpec FES and Group=TempSpec Group)
alive:=TempSpec alive.
end
--You may be allowed to delete a record, if your security level is
--higher than allowance level and the spec. is not in use.
if current status=4 then
if current user level<="3" and temp Modify="Y" then
delete records in Mechproperty with(FES+TempSpec FES and Group=TempSpec Group)
else
message"Deleting records is not valid."window.
end
exit
end
--you may be allowed to modify the spec. if "Modify" is "Y".
if temp Modify = "Y" then
modify records in Mechproperty with(FES=TempSpec FES and Group=TempSpec Group)
copy all from TempSpec;
Name:=current user name;  Date:=current date;  alive:=no.
--If user level is higher than "Medium 3" and alive is "Yes", make the spec. alive and
--copy the chemical range into "Standard --Spec." file if it has not been created before.
if current user level<="3" and TempSpec alive=Yes then
modify records in Mechproperty
alive:=TempSpec alive.
for S-Spec2 with(FES=TempSpec FES)
for Standard Spec. with(FES=TempSpec FES)
message"The chemical range for this Spec. is already existed."window.
assign temp Enter:="N".
end
if temp Enter not="N" then
```

```
enter a record in Standard Spec.
copy all from S-Spec2.
end
end
if current status=3 then
delete records in S-Spec2 with(FES=TempSpec FES)
exit
end
end
--This "End" is corresponding to "if temp Modify not="N" then".
end
--This "End" is corresponding to "if current status=2 then".
end
End.
```

Procedure 3: ***Enterspec***

Functions: This processing procedure is designed to enter Specifications into Mechanical Propertity file, Mechproperty.

```
DQL QUERY
define temp "Coefficient1" Number. define temp "Coefficient2" Number.
define global "Unit1"Text 12. define global "Unit2"Text 10.
define global "FES"Numeric String 4. define global "Source"Text 7.
define global "Spec"Text 15. define global "Grade"Text 10.
define global "Issue"Text 7. define global "Exit"Text 1.
--To assign different values to Coefficients depending on the Units chosen.
case (global Unit1)
value ("MPa" or N/Squ.mm" or "MN/Squ.m") :
assign temp Coefficient1:=1.
value ("tonf/Squ.in"): assign temp Coefficient1:=15.44.
value ("1bf/Squ.in"): assign temp Coefficient1:=0.006895.
value ("Kgf/Squ.mm"): assign temp Coefficient1:=9.807.
end
case (global Unit2)
value ("J")or ("N m"): assign temp Coefficient2:=1.
value ("ft 1bf): assign temp Coefficient2:=1.3558.
value ("Kgf m"): assign temp Coefficient2:=9.8066.
end
--Enter the Following data into Mechproperty File.
enter a record in Mechproperty
FES:=global FES; Group:="A"; alive:=no;
Date:=current date; Name:=current user name.
enter a record in S-Spec2
FES:=global FES; group:="A"; Source:=global Source; Specification:=global Spec;
Issue No.:=global Issue; Grade:=global Grade.
message "Press SH-F3 once Mechproperty is Loaded.".
input using Mechproperty into "TempSpec1".
if current status=1 or current status=4 or TempSpec1 Group not="A" then
message"This spec. is deleted, because|
you have pressed F7, Esc."window.
delete records in Mechproperty with(FES=global FES and Group="A")
delete records in S-Spsc2 with(FES=global FES)
assign global Exit:="Y".
exit
else
```

```
for Mechproperty with(FES=global FES and Group="A" and group=TempSpec1 Group);
modify records
copy all from TempSpec1;
stress1Min:=TempSpec1 Stress1Min*temp Coefficient1;
stress1Max:=TempSpec1 Stress1Max*temp Coefficient1;
Rm1Min:=TempSpec1 Rm1Min*temp Coefficient1;
Rm1Max:=TempSpec1 Rm1Max*temp Coefficient1;
stress2Min:=TempSpec1 Stress2Min*temp Coefficient1;
stress2Max:=TempSpec1 Stress2Max*temp Coefficient1;
Rm2Min:=TempSpec2 Rm2Min*temp Coefficient1;
Rm2Max:=TempSpec2 Rm2Max*temp Coefficient1;
stress3Min:=TempSpec1 Stress3Min*temp Coefficient1;
stress3Max:=TempSpec1 Stress3Max*temp Coefficient1;
Rm3Min:=TempSpec1 Rm3Min*temp Coefficient1;
Rm3Max:=TempSpec1 Rm3Max*temp Coefficient1;
stress4Min:=TempSpec1 Stress4Min*temp Coefficient1;
stress4Max:=TempSpec1 Stress4Max*temp Coefficient1;
Rm4Min:=TempSpec1 Rm4Min * temp Coefficient1;
Rm4Max:=TempSpec1 Rm4Max * temp Coefficient1;
Value1:=TempSpec1 Value1 * temp Coefficient2;
Value2:=TempSpec1 Value2 * temp Coefficient2;
Value3:=TempSpec1 Value3 * temp Coefficient2;
Value4:=TempSpec1 Value4 * temp Coefficient2;
--Modify the values,"0" which indicate the default value.
--Assign the default value with "Blank".
if Stress1Min=0 then modify records  Stress1Min:=blank. end
if Stress1Max=0 then modify records  Stress1Max:=blank. end
if Stress2Min=0 then modify records  Stress2Min:=blank. end
if Stress2Max=0 then modify records  Stress2Max:=blank. end
if Stress3Min=0 then modify records  Stress3Min:=blank. end
if Stress3Max=0 then modify records  Stress3Max:=blank. end
if Stress4Min=0 then modify records  Stress4Min:=blank. end
if Stress4Max=0 then modify records  Stress4Max:=blank. end
if Rm1Min=0 then modify records  Rm1Min:=blank. end
if Rm1Max=0 then modify records  Rm1Max:=blank. end
if Rm2Min=0 then modify records  Rm2Min:=blank. end
if Rm2Max=0 then modify records  Rm2Max:=blank. end
if Rm3Min=0 then modify records  Rm3Min:=blank. end
if Rm3Max=0 then modify records  Rm3Max:=blank. end
if Rm4Min=0 then modify records  Rm4Min:=blank. end
if Rm4Max=0 then modify records  Rm4Max:=blank. end
if Value1=0 then modify records  Value1:=blank. end
if Value2=0 then modify records  Value2:=blank. end
if Value3=0 then modify records  Value3:=blank. end
if Value4=0 then modify records  Value4:=blank. end
end
End.
```

Procedure 4: ***Modispec***

Functions: This processing procedure is designed to let user access and edit Mechanical Property records.

DQL QUERY

```
define global "Source"Text 7.  define global "Spec"Text 15.
define global "Issue"Text 7.  define global "Grade"Text 10.
```

```
define global "PreGrade"Text 10. define global "PreIssue"Text 7.
define global "FES"Numeric String 4. define global "Exit"Text 1.
for Standard Spec. with Source=global Source and Specification=global Spec. and
                    Issue No.=global preIssue and Grade=global preGrade;
delete records in S-Spec2 with (FES=Standard Spec. FES)
enter a record in S-Spec2
copy all from Standard Spec.;
FES:=global FES; Grade:=global Grade; Issue No. :=global Issue.
enter a record in Mechproperty
FES:=global FES; Group:="A"; alive:=no; name:=current user name;
Date:=current date.
end
message "Press F3 once Mechproperty File Is Loaded.".
 input using Mechproperty into"Tempspec1".
 if current status=1 or current status=4 then
message"This spec. has been delete because|
        you have pressed F7, Esc." window.
delete records in Mechproperty with (FES=global FES and group="A"
delete records in S-Spec2 with (FES=global FES)
assign global Exit:="Y".
exit
else
modify records in Mechproperty with (FES=Tempspec1 FES and Group="A")
copy all from Tempspec1.
End.
```

4-2-2 Document Distribution

4-2-2-1 Enter a Record for Document Approval from Customer

Procedure 1: ***Docu_Approval***

Functions: This is a command procedure for creating a new status record for a document that has been sent to customer. This procedure can check the document name and its number with existing records to inform user if they have already been entered. If they exist, the procedure will load the file for user to choose and edit. If they don't exist, the procedure will run a procedure, "Create_Docu._Sent" to create a new record for them and load the data file for user to enter other information.

DQL QUERY

```
define global "Document"Text 21. define global "Docu_No."Text 13.
assign global Document:=data-entry Document. assign global Docu_No.:=data-entry Docu_No..
if data-entry Are you sure? = Yes then
if firstw(data-entry Text,1)= "Create" then
run procedure"Create_Docu._Sent".
message"press SH-F3 Once the datafile is loaded.".
record entry "Docu_Sent".
else
message"Search using ALT-F5, SHFT-F3 or SHFT-F1.".
record entry "Docu_Sent".
end
End.
```

Procedure 2: *Create_Docu._Sent*

Functions: This is a processing procedure for creating a new record with key fields before allowing user to access it and enter other information.

DQL QUERY

```
define global "Document"Text 21.  define global "Docu_No."Text 13.
enter a record in Docu._Sent
Document := global Document;  Docu_No. := global Docu_No.;
date:=current date.
```

4-2-2-2 Enter a Record for internal Document Distribution

Procedure: *Docu._Distribution*

Functions: The procedure is designed to create a new status record for a document distributed at points of use within the Company. The procedure can check user's request to inform the user whether or not the record has already existed in the database to avoid duplications in the system. This procedure is similar to procedure, Create_Docu._Sent, listed in section 4-2-2-1.

4-2-2-3 Delete None Distributed Records

Procedure: *Delete_None_Distr.*

Functions: This procedure is designed to allow none distributed records to be deleted. It requires user to empty all fields for which the record is to be deleted.

4-2-3 Listing Docu. Distributions

4-2-3-1 Documents Awaiting Customer Approval

Procedure: *List_Docu._Approval*

(It is similar to procedure, List_Docu._Sent, listed in section 4-2-3-3, except that the sorting is limited to "Approval".)

4-2-3-2 Rejected Document Awaiting Re-Issue

Procedure: *List_Docu._Reissue*

(It is similar to procedure, List_Docu._Sent, listed in section 4-2-3-3, except that the sorting is limited to "Reissue".)

4-2-3-3 Documents Sent to Customers

Procedure 1: *List_Docu_Sent*

Functions: This procedure is designed to list all documents status records that have been sent to customers.

DQL QUERY

```
define temp "No."Text 25.  define temp "Docu."Text 25.  define temp "T_P"Number.
for Docu._Sent with FEL Order No.=data-entry FEL Order No.;
assign temp T_P:=temp T_P+1.
end
for Docu._Sent with FEL Order No.=data-entry FEL Order No. ;
 assign temp Docu. := Document.
```

```
assign temp No. := Docu_No..
for Approve_Issue with(Document=temp Docu. and Docu_No.=temp No.)
list records
data-entry FEL Order No.; temp T_P; Document in groups; Docu_No.;
Issue; Date_Sent; Approval; Date_Approved; comment.
end.
End.
```

4-2-3-4 List Internal Distributions Not yet returned

Procedure: *Print_NRD_Report*

Functions: This procedure is designed to list all departments that have not returned previous issues of specified process plan.

4-2-3-5 List Document Distribution for Specified Department

Procedure: *List_QA._Document*

Functions: This procedure is designed to list a specific department which is holding un-returned issues of process plans.

4-2-4 FEL Melting Programmes

4-2-4-1 Check FSL Melting Programme

Procedure 1: *Checkprogramme*

Functions: This is a command procedure designed to compare FSL Melting Programme with FEL Steel Order Requirements and to compare FSL Ingot Detail with FEL Ingot Requirements. Any difference will be displayed on the screen for users.

```
DQL QUERY
define global "Item"Numeric String 5. define global "Issue"Text 2. define global "ST_No."Text 6.
define global "Version"Text 1. define global "R0"Text 3. define global "R1"Text 3 .
define global "R2"Text 3. define global "BIG"Text 1 .
assign global ITEM:=data-entry ITEM. assign global ISSUE:=data-entry ISSUE.
define global "Message1"Text 40. define global "Message2"Text 40.
define global "Message3"Text 40. define global "Message4"Text 40.
define global "Message5"Text 40. define global "Message6"Text 40.
define global "Message7"Text 40. define global "Message8"Text 40.
define global "Spare Metal"Text 4. define global "Nozzle Size"Text 4.
define global "Aiming Temp."Text 4. define global "Spray Limiter"Text 4.
define global "Head Filled"Text 5. define global "Process"Text 34.
define global "Deoxidisation"Text 20. define global "Hot to Forge"Text 3.
run procedure "Comparmelting".
run procedure "Comparingot".
if global BIG="Y" then
run Procedure "Comparingotdetail".
end
if global R2="not" and data-entry Create = Yes then
run procedure "Creatediscrepancy".
message"Discrepancy Report has been created. Edit it please."window.
End.
```

Procedure 2: ***Comparmelting***
(Same as procedure 4 listed in section 4-2-4-2)

Procedure 3: ***Comparingot***
(Same as procedure 5 listed in section 4-2-4-2)

Procedure 4: ***Comparingotdetail***

Functions: This is a processing procedure for diagnose FSL steel making procedures for ingots over a specific diameter. The checking process is done through matching rules in data file, FSLbigingot.

DQL QUERY

```
define global "ITEM"Numeric String 5. define global "ISSUE"Text 2.
define global "ST_No."text 6. assign global "Version"text 1.
assign global "Message1"Text 40. assign global "Message2"Text 40.
assign global "Message3"Text 40. assign global "Message4"Text 40.
assign global "Message5"Text 40. assign global "Message6"Text 40.
assign global "Message7"Text 40. assign global "Message8"Text 40.
assign global "Spare Metal"text 4. assign global "Nozzle Size"text 4.
assign global "Aiming Temp."text 4. define global "Spray Limiter"Text 4.
define global "Head Filled"Text 5. define global "Process"Text 34.
define global "Deoxidisation"Text 20. define global "Hot To Forge"Text 3.
define temp "Cond1"text 3. define temp "Cond2"text 3.
define temp "Cond3"text 3. define global "R"Text 3.
for FSLbigingot with ITEM=global ITEM and ISSUE=global ISSUE;
for Steelord with(ST_No.=global ST_No. and Version=global Version)
for FEL Range with(Melting No.=Steelord Melting No.)
for FSLingot with ITEM=global ITEM and ISSUE=global ISSUE and INGOTS>="Igt_X M/F")
for Ingots with(ST_No.=global ST_No. and Version=global Version and
                Ingot Size = FSLingot INGOTS)
--VCD Requirement or [S] and [P] content <=0.008%
if Steelord Deoxidisation=VCD or FEL Range SMax<=0.008 or FEL Range PMax<=0.008 then
  assign temp Cond1:="Yes".
else
  assign temp Cond1:="N".
end
if Steelord Deoxidation=A1 then
  assign temp Cond2:="Yes".
else
  assign temp Cond2:="no".
end
if FEL Range Quality="1 CrMoV" then
  assign temp Cond3:="Yes".
else
  assign temp Cond3:="no".
end
--Compare 8 Main Parameter in Melting Programme detail for ingot size over a specific diameter.
for Standard72 with(A1 TREATED=temp Cond2 and 1 CrMoV=temp Cond3)
--1. Nozzle Size:
if FSLbigingot NOZZLE SIZE not=NOZZLE SIZE then
assign global Message2:="The Nozzle Size is not Standard.".
assign global R2:="not".
else
assign global Message2:="The Nozzle Size is OK.".
```

```
end
assign global Nozzle Size:=FSLbigingot NOZZLE SIZE.
--2. Aiming Temp.:
if FSLbigingot AIM TEMP. not>=FSLbigingot LIQUIDUS+T_PLUS then
assign global Message3:="The AIM Temp. is Lower than Standard.".
assign global R2:="not".
else
assign global Message3:="The Aim Temp. is OK.".
end
assign global Aiming Temp.:=FSLbigingot AIM TEMP..
--3. Spray Limiter:
if FSLbigingot spray not=SPRAY LIMITER then
assign global Message4:="The New Spray Limiter is not Applied.".
assign global R2:="not".
else
assign global Message4:="The New Spray Limiter is Applied.".
end
assign global Spray Limiter:=FSLbigingot Spray.
--4. Head Filled:
if FSLbigingot HEAD FILLED not=HEAD FILLED To then
assign global Message5:="The Head Filled is not Sufficient.".
assign global R2:="not".
else
assign global Message5:="The Ingot Head Filled is OK.".
end
assign global Head Filled:=FSLbigingot HEAD FILLED.
--5. Liquid Metal Weight:
if FSLbigingot LIQUID WT.<Ingots Weight+SPARE then
assign global Message1:="The Spare Metal is not sufficient.".
assign global R2:="not".
else
assign global Message1:="The Actual Spare Metal Weight is OK.".
end
assign global Spare Metal:=FSLbigingot LIQUID WT.-Ingots Weight.
--6. Process Route
for process with(ID.=Standard72 ID. and VCD TREAED = temp Cond1 and
        WEIGHT1<=Ingots Weight and WEIGHT2>=Ingots Weight)
if PROCESS1 not=FSLbigingot PROCESS ROUTE 1 or PROCESS2 not=
                FSLbigingot PROCESS ROUTE 2 then
assign global Message6:="The Process is not a Proper Route."
assign global R2:="not".
else
assign global Message6:="The Process Route is OK.".
end
assign global Process:=jointext(firstw(FSLbigingot PROCESS ROUTE 1,2),":").
assign global Process:=jointext(global Process, FSLbigingot PROCESS ROUTE 2).
end
--7. Deoxidisation:
if Steelord Deoxidisation not=Not Specified then
if Steelord Deoxidisation not=FSLbigingot DEOXIDISATION 2 then
assign global Message7:="The Deoxidisation is not same to FEL.".
assign global R2:="not".
else
assign global Message7:="The Deoxidisation is OK.".
end
else
```

```
assign global Message7:="The Deoxidisation is OK.".
end
assign global Deoxidisation:=jointext(FSLbigingot DEOXIDISATION 1," * ").
assign global Deoxidisation:=jointext(global DEOXIDISATION, FSLbigingot DEOXIDISATION 2).
--8. Hot To Forge:
if FSLbigingot HOT TO FORGE not=Steelord Hot To Forge then
assign global Message8:="Hot To Forge Is Not Specified.".
else
assign global Message8:="Hot To Forge is Specified.".
end
assign global Hot To Forge:=FSLbigingot HOT TO FORGE.
if global R"="not" then
list records
 Steelord ST_No.;  Steelord Version;
  FSLbigingot ITEM;  FSLbigingot ISSUE;
  global Message1;global Spare Metal;  global Message2;global Nozzle Size;
  global Message3;global Aiming Temp;  global Message4;global Spray Limiter;
  global Message5;global Head Filled;  global Message6;global Process;
  global Message7;global Deoxidisation;  global Message8;global Hot To Forge;
  global R2.
else
message"50,10,10,7,FSL Steelmaking Program Satisfies the following conditions:|
                        Spare Metal and Nozzle Size;|
                        Metal Processes and Aiming Temp.;|
                        Head Filled and Spray Limiter;|
                        Deoxidisation and Delivery;"window.
end end end  end  end
End.
```

Procedure 5: ***Creatediscrepancy***

Functions: This processing procedure is designed to create a discrepancy report by saving all checking results generated by previous processing procedures.

```
DQL QUERY
define global "R0"Text 3.  define global "R1"Text 3.  define global "R2"Text 3.
assign global "ITEM"Numeric String 5.  assign global "ISSUE"Text 2.
define global "Message1"Text 40.  define global "Message2"Text 40.
define global "Message3"Text 40.  define global "Message4"Text 40.
define global "Message5"Text 40.  define global "Message6"Text 40.
define global "Message7"Text 40.  define global "Message8"Text 40.
define global "Spare Metal"Text 4.  define global "Nozzle Size"Text 4.
define global "Aiming Temp."Text 4.  define global "Spray Limiter"Text 4.
define global "Head Filled"Text 5.  define global "Process"Text 34.
define global "Deoxidisation"Text 20.  define global "Hot to Forge"Text 3.
define temp "Modify"Text 3.
for Discrepancy with (Item=global ITEM and Issue=global ISSUE);
--If the record exists with identify of  Item number and its Issue which are equal to global variables,
--ITEM and ISSUE, the temp Modify will be assigned a value of "Yes".
assign temp Modify := "Yes".
modify records
 Message1 := global Message1;  value1 := global Spare Metal;
 Message2 := global Message2;  value2 := global Nozzle Size;
 Message3 := global Message3;  value3 := global Aiming Temp.;
 Message4 := global Message4;  value4 := global Spray Limiter;
 Message5 := global Message5;  value5 := global Head Filled;
```

```
  Message6 := global Message6;  value6 := global Process;
  Message7 := global Message7;  value7 := global Deoxidisation;
  Message8 := global Message8;  value8 := global Hot to Forge.
end
if temp Modify not= "Yes" then
enter a record in Discrepancy1
  Item := global ITEM;  Issue := global;  Date := current date;
  Message1 := global Message1;  Message2 := global Message2;  Message3 := global Message3;
  Message4 := global Message4;  Message5 := global Message5;  Message6 := global Message6;
  Message7 := global Message7;  Message8 := global Message8;  value1 := global Spare Metal;
  value2 := global Nozzle Size;  value3 := global Aiming Temp.;  value4 := global Spray Limiter;
  value5 := global Head Filled;  value6 := global Process;  value7 := global Deoxidisation;
  value8 := global Hot to Forge.
end
--if variable "R0" equals "No", that means the Ingot Requirements are not acceptable.
if global R0 ="No" then
modify records in Discrepancy1 with (Item = global ITEM and Issue = global ISSUE)
 Ingot Requirement:=No.
else
modify records in Discrepancy1 with (Item = global ITEM and Issue = global ISSUE)
Ingot Requirement:=Yes.
end
if global R1 ="Out" then
modify records in Discrepancy1 with (Item = global ITEM and Issue = global ISSUE)
Melting Range:=No.
else
modify records in Discrepancy1 with (Item = global ITEM and Issue = global ISSUE)
Melting Range:=Yes.
End.
```

4-2-4-2 Melting programme Data-Entry

Procedure 1: *Melting Programme*

Functions: This is a command procedure for controlling the data entry process of FSL Steelmaking Procedures. It can check user's request with the database files to avoid duplicated records. It can also transfer information such as Item No. and Issue into Forging Status file to create the link with the Forging Definition. It then allows user to access to the datafile, FSL Proc. to enter other information from FSL melting procedure. It then runs two processing procedures, "Comparmelting" and "Comparingot", to compare FEL melting range with FSL designed melting range and to compare FEL ingot design parameters with FSL design parameters, and finally list any difference for user.

```
DQL QUERY
define global "ITEM"Numeric String 5.  define global "Issue"Text 2.
define global "ST_No."Text 6.  define global "Version"Text 1.
define global "R0"Text 3.  define global "R1"Text 3.
define global "R2"Text 3.  define global "BIG INGOT "Text 3.
define global "FEL Order No."Text 11.  define global "FEL Order 1"Numeric String 7.
define global "FEL Order 2"Numeric String 7.  define global "FEL Order 3"Numeric String 7.
define global "FEL Order 4"Numeric String 7.  define global "FEL Order 5"Numeric String 7.
define global "FEL Order No. 1"Numeric String 7.  define global "Number"Number.
assign global FEL Order No.:=data-entry FEL Order No..
assign global FEL Order 1:=data-entry FEL Order 1.
assign global FEL Order 2:=data-entry FEL Order 2.
```

```
assign global FEL Order 3:=data-entry FEL Order 3.
assign global FEL Order 4:=data-entry FEL Order 4.
assign global ITEM:=data-entry ITEM.
assign global ISSUE:=data-entry ISSUE.
if data-entry Are you sure? = Yes then
--"Postdata3" is for assigning ITEM and ISSUE No. to each FEL Order record.
if data-entry Big ingot = Yes then
assign global BIG INGOT :="Y".
end
run procedure "Postdata3".
message "Press SH-F3 Once FSL Proc Form Is Loaded.".
record entry "FSLproc".
run procedure "Comparmelting".
run procedure "Comparingot".
if data-entry Big ingot = Yes then
record entry "FSLbigingot".
end
end.
```

Procedure 2: Postdata3

Functions: This is a processing procedure for posting FSL Item No. and it's Issue No. into Forging Status File, and creating a record with these identities in data files, FSL Proc and FSL Ingot.

```
DQL QUERY
define temp "ST_No."Text 6. define temp "ST_I "Text 1.
define global "FEL Order No."text 11. define global "FEL Order No. 1"Numeric String 7.
define global "FEL Order 1"Numeric String 7. define global "FEL Order 2"Numeric String 7.
define global "FEL Order 3"Numeric String 7. define global "FEL Order 4"Numeric String 7.
define global "FEL Order 5"Numeric String 7. define global "Item"Numeric String 5.
define global "Issue"text 2. define global "Cast"text 10.
define global "Letter"text 5. define global "Big ingot"Text 3.
define temp "first No."Numeric String 7. define temp "last No."Numeric String 7.
define temp "firsthalf No."Number. define global "Number"Number.
define temp "N"Number. define temp "N1"Number.
define temp "N2"Number.
if length(global FEL Order No.) >=9 then
case(length (global FEL Order No.))
value (9):
assign temp N1:=1. assign temp N2:=6.
value (10):
assign temp N1:=2. assign temp N2:=5.
value (11):
assign temp N1:=3. assign temp N2:=4.
end
assign global Number:=temp last No. - temp first No.+1.
assign temp N:=global Number.
else
assign global Number:=1. assign temp N:=1.
end
assign temp first No.:=firstc(global FEL Order No.,7).
assign global FEL Order No.1:=temp first No..
assign temp firsthalf No.:=firstc(global FEL Order No.,temp N2).
assign temp last No.:=jointext(temp firsthalf No., last(global FEL Order No.,(temp N1))).
while temp N>=1 do
```

```
 for SO_Status with(FEL Order No.=temp first No.);
  assign temp ST_No.:=ST_No..  assign temp ST_I:=Version.
  if global Item not=blank then
  modify records
  Item:=global Item;  Issue:=global Issue.
  else
  if global Cast not=blank then
  modify records
  Cast No.:=global Cast;  Letter:=global Letter.
  end   end   end
 assign temp first No.:=temp first No.+1.
 assign temp N:=temp N - 1.
 end
 if FEL Order 1 not=blank then
 for SO_status with(FEL Order No.=FEL Order  1 or FEL Order No.=FEL Order 2 or FEL Order No.=
                    FEL Order 3 or FEL Order No.=FEL Order 4 or FEL Order No.=FEL Order 5);
 if global Item not=blank then
 modify records
 Item:=global Item;  Issue:=global Issue.
 else
 if global Cast not=blank then
 modify records
 cast No.:=global Cast;  Letter:=global Letter.
 end  end
   assign global Number:=global Number+1.
   end    end
  if global Item not=blank then
  for Issue_Steelord with(ST_No.=temp ST_No. and Version=temp ST_I);
  for FEL Range
  enter a record in FSLproc
  ITEM:=global Item;  ISSUE:=global Issue;
  Date:=current date;  Ded1:=FEL Range Ded1;
  Ded2:=FEL Range Ded2;  Ded3:=FEL Range Ded3;
  Ded4:=FEL Range Ded4.
  end
  end
  if global Big ingot = "Y" then
  enter a record in FSLbigingot
  ITEM:=global Item;  ISSUE:=global Issue.
  end
 else
 -- copy Cast number into Chemical Analysis File.
   if global Cast not=blank then
    for Issue_Steelord with(ST_No.=temp ST_No. and Version=ST_I);
    for FEL Range
    enter a record in Analysis_Index
    Cast:=jointext(global Cast,global Letter);
    Date:=current date;  Ded1:=FEL Range Ded1;
    Ded2:=FEL Range Ded2;  Ded3:=FEL Range Ded3;
    Ded4:=FEL Range Ded4.
    end
    end
  end
 End.
```

Procedure 3: *FSLproc*

Functions: This procedure s designed to allow user to access and enter FSL Steelmaking Programme into a data file, FSL Proc, including FSL chemical range.

Procedure 4: *Comparmelting*

Functions: This processing procedure is designed to compare FEL steel melting range with FSL designed melting range and list any difference for user.

```
DQL QUERY
define global "Item"Numeric String 5. define global "Issue"text 2.
define global "R1"Text 5. define temp "Quality"Text 15. define temp "RMC"Text 13.
define temp "CMin"Text 6. define temp "CMax"Text 6. define temp "SiMin"text 6.
define temp "SiMax"text 6. define temp "MnMin"Text 6. define temp "MnMax"Text 6.
define temp "PMin"Text 6. define temp "PMax"Text 6. define temp "SMin"text 6.
define temp "SMax"text 6. define temp "CrMin"Text 6. define temp "CrMax"Text 6.
define temp "MoMin"Text 6. define temp "MoMax"Text 6. define temp "NiMin"Text 6.
define temp "NiMax"Text 6. define temp "VMin"text 6. define temp "VMax"text 6.
define temp "AlMin"Text 6. define temp "AlMax"Text 6. define temp "SbMin"Text 6.
define temp "SbMax"text 6. define temp "SnMin"Text 6. define temp "SnMax"Text 6.
define temp "CuMin"Text 6. define temp "CuMax"Text 6. define temp "NMin"text 6.
define temp "NMax"text 6. define temp "AsMin"Text 6. define temp "AsMax"Text 6.
define temp "NbMin"Text 6. define temp "NbMax"Text 6. define temp "TiMin"Text 6.
define temp "TiMax"Text 6. define temp "Ded1Min"text 6. define temp "Ded1Max"text 6.
define temp "Ded2Min"Text 6. define temp "Ded2Max"Text 6. define temp "Ded3Min"Text 6.
define temp "Ded3Max"text 6. define temp "Ded4Min"Text 6. define temp "Ded4Max"Text 6.
define temp "H2Min"Text 6. define temp "H2Max"text 6. define global "ST_No."text 6.
define global "Version"text 1. define temp "RMC1"text 8.
for FSLproc with(ITEM=global Item and ISSUE=global Issue);
for Status with(ITEM=global Item and ISSUE=global Issue);
assign global ST_No.:=ST_No.. assign global Version:=ST_I.
 for Manuf. plan
 assign temp RMC1:=RMC.
 end
 break
end
for Steelord with(ST_No.=global ST_No. and Version=global Version)
for FEL Range with(Melting No.=Steelord Melting No.)
--Check Melting Ranges and assign differences to temp. variables for listing.
if QUALITY not=FSLproc QUALITY then
assign global R1:="Out". assign temp Quality:=FSLproc QUALITY.
else
assign temp Quality:=(Same to  FEL)".
end
if temp RMC1 not=FSLproc R.M.C. then
assign global R1:="Out". assign temp RMC:=FSLproc R.M.C.
else
assign temp RMC:=(Same to  FEL)".
end
if CMin>FSLproc CMin or CMax<FSLproc CMin then
assign global R1:="Out". assign temp CMin:=FSLproc CMin.
assign temp CMax:=FSLproc CMax.
end
if SiMin>FSLproc SiMin or SiMax<FSLproc SiMin then
assign global R1:="Out". assign temp SiMin:=FSLproc SiMin.
assign temp SiMax:=FSLproc SiMax.
```

```
end
if MnMin =blank and FSLproc MnMin not=blank then
assign global R1:="Out". assign temp MnMin:=FSLproc MnMin.
end
if MnMin>FSLproc MnMin or MnMax<FSLproc MnMax then
assign global R1:="Out". assign temp MnMin:=FSLproc MnMin.
assign temp MnMax:=FSLproc MnMax.
end
if PMin =blank and FSLproc PMin not=blank then
assign global R1:="Out". assign temp PMin:=FSLproc PMin.
end
if PMin>FSLproc PMin or PMax<FSLproc PMax then
assign global R1:="Out". assign temp PMin:=FSLproc PMin.
assign temp PMax:=FSLproc PMax.
end
if SMin =blank and FSLproc SMin not=blank then
assign global R1:="Out". assign temp SMin:=FSLproc SMin.
end
if SMin>FSLproc SMin or SMax<FSLproc SMax then
assign global R1:="Out". assign temp SMin:=FSLproc SMin.
assign temp SMax:=FSLproc SMax.
end
if CrMin =blank and FSLproc CrMin not=blank then
assign global R1:="Out". assign temp CrMin:=FSLproc CrMin.
end
if CrMin>FSLproc CrMin or CrMax<FSLproc CrMax then
assign global R1:="Out". assign temp CrMin:=FSLproc CrMin.
assign temp CrMax:=FSLproc CrMax.
end
if MoMin =blank and FSLproc MoMin not=blank then
assign global R1:="Out". assign temp MoMin:=FSLproc MoMin.
end
if MoMin>FSLproc MoMin or MoMax<FSLproc MoMax then
assign global R1:="Out". assign temp MoMin:=FSLproc MoMin.
assign temp MoMax:=FSLproc MoMax.
end
if NiMin =blank and FSLproc NiMin not=blank then
assign global R1:="Out". assign temp NiMin:=FSLproc NiMin.
end
if NiMin>FSLproc NiMin or NiMax<FSLproc NiMax then
assign global R1:="Out". assign temp NiMin:=FSLproc NiMin.
assign temp NiMax:=FSLproc NiMax.
end
if VMin =blank and FSLproc VMin not=blank then
assign global R1:="Out". assign temp VMin:=FSLproc VMin.
end
if VMin>FSLproc VMin or VMax<FSLproc VMax then
assign global R1:="Out". assign temp VMin:=FSLproc VMin.
assign temp VMax:=FSLproc VMax.
end
if AlMin =blank and FSLproc AlMin not=blank then
assign global R1:="Out". assign temp AlMin:=FSLproc AlMin.
end
if AlMin>FSLproc AlMin or AlMax<FSLproc AlMax then
assign global R1:="Out". assign temp AlMin:=FSLproc AlMin.
assign temp AlMax:=FSLproc AlMax.
```

```
end
if SbMin =blank and FSLproc SbMin not=blank then
assign global R1:="Out".  assign temp SbMin:=FSLproc SbMin.
end
if SbMin>FSLproc SbMin or SbMax<FSLproc SbMax then
assign global R1:="Out".  assign temp SbMin:=FSLproc SbMin.
assign temp SbMax:=FSLproc SbMax.
end
if SnMin =blank and FSLproc SnMin not=blank then
assign global R1:="Out".  assign temp SnMin:=FSLproc SnMin.
end
if SnMin>FSLproc SnMin or SnMax<FSLproc SnMax then
assign global R1:="Out".  assign temp SnMin:=FSLproc SnMin.
assign temp SnMax:=FSLproc SnMax.
end
if CuMin =blank and FSLproc CuMin not=blank then
assign global R1:="Out".  assign temp CuMin:=FSLproc CuMin.
end
if CuMin>FSLproc CuMin or CuMax<FSLproc CuMax then
assign global R1:="Out".  assign temp CuMin:=FSLproc CuMin.
assign temp CuMax:=FSLproc CuMax.
end
if NMin =blank and FSLproc NMin not=blank then
assign global R1:="Out".  assign temp NMin:=FSLproc NMin.
end
if NMin>FSLproc NMin or NMax<FSLproc NMax then
assign global R1:="Out".  assign temp NMin:=FSLproc NMin.
assign temp NMax:=FSLproc NMax.
end
if AsMin =blank and FSLproc AsMin not=blank then
assign global R1:="Out".  assign temp AsMin:=FSLproc AsMin.
end
if AsMin>FSLproc AsMin or
  AsMax<FSLproc AsMax then
assign global R1:="Out".  assign temp AsMin:=FSLproc AsMin.
assign temp AsMax:=FSLproc AsMax.
end
if NbMin =blank and FSLproc NbMin not=blank then
assign global R1:="Out".  assign temp NbMin:=FSLproc NbMin.
end
if NbMin>FSLproc NbMin or NbMax<FSLproc NbMax then
assign global R1:="Out".  assign temp NbMin:=FSLproc NbMin.
assign temp NbMax:=FSLproc NbMax.
end
if TiMin =blank and FSLproc TiMin not=blank then
assign global R1:="Out".  assign temp TiMin:=FSLproc TiMin.
end
if TiMin>FSLproc TiMin or TiMax<FSLproc TiMax then
assign global R1:="Out".  assign temp TiMin:=FSLproc TiMin.
assign temp TiMax:=FSLproc TiMax.
end
if Ded1Min =blank and FSLproc Ded1Min not=blank then
assign global R1:="Out".  assign temp Ded1Min:=FSLproc Ded1Min.
end
if Ded1Min>FSLproc Ded1Min or Ded1Max<FSLproc Ded1Max then
assign global R1:="Out".  assign temp Ded1Min:=FSLproc Ded1Min.
```

```
assign temp Ded1Max:=FSLproc Ded1Max.
end
if Ded2Min =blank and FSLproc Ded2Min not=blank then
assign global R1:="Out".  assign temp Ded2Min:=FSLproc Ded2Min.
end
if Ded2Min>FSLproc Ded2Min or Ded2Max<FSLproc Ded2Max then
assign global R1:="Out".  assign temp Ded2Min:=FSLproc Ded2Min.
assign temp Ded2Max:=FSLproc Ded2Max.
end
if Ded3Min =blank and FSLproc Ded3Min not=blank then
assign global R1:="Out".  assign temp Ded3Min:=FSLproc Ded3Min.
end
if Ded3Min>FSLproc Ded3Min or Ded3Max<FSLproc Ded3Max then
assign global R1:="Out".  assign temp Ded3Min:=FSLproc Ded3Min.
assign temp Ded3Max:=FSLproc Ded3Max.
end
if Ded4Min =blank and FSLproc Ded4Min not=blank then
assign global R1:="Out".  assign temp Ded4Min:=FSLproc Ded4Min.
end
if Ded4Min>FSLproc Ded4Min or Ded4Max<FSLproc Ded4Max then
assign global R1:="Out".  assign temp Ded4Min:=FSLproc Ded4Min.
assign temp Ded4Max:=FSLproc Ded4Max.
end
if H2Min =blank and FSLproc H2Min not=blank then
assign global R1:="Out".  assign temp H2Min:=FSLproc H2Min.
end
if H2Min>FSLproc H2Min or H2Max<FSLproc H2Max then
assign global R1:="Out".  assign temp H2Min:=FSLproc H2Min.
assign temp H2Max:=FSLproc H2Max.
end
--List full FEL Melting Range and differences from FSL designed melting range
--if global variable "R" is "out".
if global R1 ="out" then
List records
FSLproc ITEM;  FELproc ISSUE;  Quality;  temp RMC1;
Steelord ST_No.;  Steelord Version;
 CMin;Simin;MnMin;PMin;SMin;CrMin;MoMin;NiMin;VMin;
 AlMin;SbMin;SnMin;CuMin;NMin;AsMin;TiMin;Ded1Min;
 Ded2Min;Ded3Min;Ded4Min;H2Min;
 Cmax;SiMax;MnMax;PMax;SMax;CrMax;MoMax;NiMax;VMax;
 AlMax;SbMax;SnMax;CuMax;NMax;AsMax;TiMax;Ded1Max;
 Ded2Max;Ded3Max;Ded4Max;H2Max;
 temp Quality;temp RMC;  Ded1;Ded2;Ded3;Ded4;
 temp CMin;temp SiMin;temp MnMin;temp PMin;
 temp SMin;temp CrMin;temp MoMin;tempNiMin;
 tempVMin;tempAlMin;temp SbMin;temp SnMin;
 temp CuMin;temp NMin;temp AsMin;temp NbMin;
 temp TiMin;temp Ded1Min;t emp Ded2Min;
 temp Ded3Min;temp Ded4Min;temp H2Min;
 temp CMax;temp SiMax;temp MnMax;temp PMax;
 temp SMax;temp CrMax;temp MoMax;temp NiMax;
 temp VMax;temp AlMax;temp SbMax;temp SnMax;
 temp CuMax;temp NMax;temp AsMax;temp NbMax;
 temp TiMax;temp Ded1Max;temp Ded2Max;temp Ded3Max;
 temp Ded4Max;temp H2Max.
else
```

```
message "40,11,16,4,(The F.S.L. Melting Range Is)|
within| [FEL Melting Range Requirements.]" window.
end end end
End.
```

Procedure 5: Comparingot

Functions: This processing procedure is designed to compare ingot design parameters between FEL and FSL. Any difference can be listed on the screen for user.

```
DQL QUERY
define global "BIG"Text 1. define global "Item"Numeric String 5.
define global "Issue"Text 2. define global "ST_No."Text 6.
define global "Version"Text 1. define global "RO"text 3 .
define temp "L"text 3. define temp "Message"text 75.
define temp "N"number. define temp "N1"number.
for FSLingot with(ITEM=global Item and ISSUE =global Issue);
assign temp L:="N". assign temp N:=temp N+1
if firstw(ingots,1)>="Igt_X" then
assign global BIG:="Y"
end
for Ingots with(ST_No.=global ST_No. and Version=global
Version and Ingot Size=FSLingot INGOTS)
assign temp N1:=temp N1+1.
for Ingot_Status with(ST_No.=ingots ST_No. and Version=Ingots Version and Letter=ingots Letter)
  if FSLingot INGOTS not=Ingots Ingot Size or
  FSLingot Len/Chiwt not=Ingots Len/Chiwt. or
  FSLingot Length not=Ingots L/C.W or
  FSLingot DEST'N not=Ingots Delivery To or
  FSLingot TIME DUE not<=Ingots TCD or
  FSLingot ORDER No. not=FEL Order No. then
assign global R0:="No".
assign temp L:="Y".
list records
global Item; global Issue; ST_No.; Version;
Ingots Ingot Size in groups;
 Ingots Len/Chiwt.; Ingots L/C.w; Ingots Quality; Ingots Weight;
 Ingots Delivery To; Ingots TCD; Ingots Cost; FEL Order No.;
 FSLingot Len/Chiwt; FSLingot ORDER No. in group;
 FSLingot LENGTH; FSLingot WEIGHT;
 FSLingot DEST'N; FSLingot TIME DUE.
else
assign global R0:="Yes".
end end end
--variable "Big" is for ingot sizes over a specific diameter, R0 for ingot not acceptable.
if temp L not="Y" and temp N=temp N1 then
message"FSL ingot design parameters are within FEL requirements."window.
else
if temp N not=temp N1 then
message"One of the ingot is not matched, Please check."window.
assign global BIG:="N".
end end
End.
```

4-2-4-3 Print Discrepancy Report

Procedure 1: ***Printdiscrepancy1***

Functions: This processing procedure is designed to print discrepancy reports.

```
DQL QUERY
for Discrepancy1 with Item = data-entry ITEM and Issue = data-entry ISSUE;
list records
 Item; Issue; Date;
 Message1; value1; Message2; value2; Message3; value3;
 Message4; value4; Message5; value5; Message6; value6;
 Message7; value7; Message8; value8; Melting Range; Ingot Requirement;
 F.E.L. Comments1; F.E.L. Comments2; F.E.L. Comments3;
 all Status with(Item-data-entry ITEM and Issue=data-entry)FEL Order No. in order.
End.
```

4-2-5 Chemical Analysis

4-2-5-1 Check Forging Cast Analysis

Procedure 1: ***Check_Analysis***

Functions: This is a command procedure for checking the cast analysis against FEL melting range specified in the steel order by calling a processing procedure, Compare_Analysis. It can also check the identify of the cast analysis record in the database to inform user whether or not the cast analysis has already existed.

```
DQL QUERY
define global "Cast" Text 12. define global "R" Text 3.
assign global CAST:=data-entry CAST.
if data-entry are you sure?=Yes then
run procedure "Compare_Analysis".
End.
```

Procedure 2: ***Compare_Analysis***

Functions: This procedure is designed to compare cast analysis with FEL melting range designed in the Steel Order data file. Any difference will be displayed on screen for users.

```
DQL QUERY
define global"Cast"Text 12. define temp"ST_No."Text 6. define temp "Version"Text 1.
define global "R" text 1. define global"Out"Text 1. define temp"C"Text 6.
define temp"Si"Text 6. define temp"Mn"Text 6. define temp"P"Text 6.
define temp"S"Text 6. define temp"Cr"Text 6. define temp"Mo"Text 6.
define temp"Ni"Text 6. define temp"V"Text 6. define temp"Al"Text 6.
define temp"Sb"Text 6. define temp"Sn"Text 6. define temp"Cu"Text 6.
define temp"N"Text 6. define temp"As"Text 6. define temp"Nb"Text 6.
define temp"Ti"Text 6. define temp"Ded1"Text 6. define temp"Ded2"Text 6.
define temp"Ded3"Text 6. define temp"Ded4"Text 6. define temp"H2"Text 6.
for SO_Status with Cast No.=firstc(global Cast,5);
assign temp ST_No.:=ST_No.. assign temp Version:=Version.
for Analysis with(Cast=global Cast)
assign global Out:=blank.
for Issue_Steelord with(ST_No.=temp ST_No. and Version=temp Version)
for FEL Range with(Melting  No.=Issue_Steelord Melting No.)
```

```
if CMax not=blank then
if Analysis C not between CMin to CMax then
assign global Out:="Y". assign temp C:=Analysis C.
end
else
if Analysis C<CMin then
assign global Out:="Y". assign temp C:=Analysis C.
else if CMin=blank then
assign global Out:="Y". assign temp C:=Analysis C.
end end end
if SiMax not=blank then
if Analysis Si not between SiMin to SiMax then
assign global Out:="Y". assign temp Si:=Analysis Si.
if Analysis Si=blank then assign temp Si:="***".
end end
else
if Analysis Si<SiMin then
assign global Out:="Y". assign temp Si:=Anlysis Si.
if Analysis Si=blank then assign temp Si:="***". end
else if SiMin=blank and Analysis Si not=blank then
assign global Out:="Y". assign temp Si:=Analysis Si.
end end end
if MnMax not=blank then
if Analysis Mn not between MnMin to MnMax then
assign global Out:="Y". assign temp Mn:=Analysis Mn.
if Analysis Mn=blank then assign temp Mn:="***".
end end
else
if Analysis Mn<MnMin then
assign global Out:="Y". assign temp Mn:=Anlysis Mn.
if Analysis Mn=blank then assign temp Mn:="***". end
else if MnMin=blank and Analysis Mn not=blank then
assign global Out:="Y". assign temp Mn:=Analysis Mn.
end end end
if PMax not=blank then
if Analysis P not between PMin to PMax then
assign global Out:="Y".
assign temp P:=Analysis P.
if Analysis P=blank then assign temp P:="***".
end end
else
if PMin>Analysis P then
assign global Out:="Y". assign temp P:=Analysis P.
if Analysis P=blank then assign temp P:="***". end
else if PMin=blank and Analysis P not=blank then
assign global Out:="Y". assign temp P:=Analysis P.
end end end
if SMax not=blank then
if Analysis S not between SMin to SMax then
assign global Out:="Y". assign temp S:=Analysis S.
if Analysis S=blank then assign temp S:="***".
end end
else
if SMin>Analysis S then
assign global Out:="Y". assign temp S:=Anlysis S.
if Analysis S=blank then assign temp S:="***". end
```

```
else if SMin=blank and Analysis S not=blank then
assign global Out:="Y". assign temp S:=Analysis S.
end end end
if CrMax not=blank then
if Analysis Cr not between CrMin to CrMax then
assign global Out:="Y". assign temp Cr:=Analysis Cr.
if Analysis Cr=blank then assign temp Cr:="***".
end end
else
if CrMin>Analysis Cr then
assign global Out:="Y". assign temp Cr:=Anlysis Cr.
if Analysis Cr=blank then assign temp Cr:="***". end
else if CrMin=blank and Analysis Cr not=blank then
assign global Out:="Y". assign temp Cr:=Analysis Cr.
end end end
if MoMax not=blank then
if Analysis Mo not between MoMin to MoMax then
assign global Out:="Y".
assign temp Mo:=Analysis Mo.
if Analysis Mo=blank then assign temp Mo:="***".
end end
else
if MoMin>Analysis Mo then
assign global Out:="Y". assign temp Mo:=Analysis Mo.
if Analysis Mo=blank then assign temp Mo:="***". end
else if MoMin=blank and Analysis Mo not=blank then
assign global Out:="Y". assign temp Mo:=Analysis Mo.
end end end
if NiMax not=blank then
if Analysis Ni not between NiMin to NiMax then
assign global Out:="Y". assign temp Ni:=Analysis Ni.
if Analysis Ni=blank then assign temp Ni:="***".
end end
else
if NiMin>Analysis Ni then
assign global Out:="Y". assign temp Ni:=Anlysis Ni.
if Analysis Ni=blank then assign temp Ni:="***". end
else if NiMin=blank and Analysis Ni not=blank then
assign global Out:="Y". assign temp Ni:=Analysis Ni.
end end end
if VMax not=blank then
if Analysis V not between VMin to VMax then
assign global Out:="Y". assign temp V:=Analysis V.
if Analysis V=blank then assign temp V:="***".
end end
else
if VMin>Analysis V then
assign global Out:="Y". assign temp V:=Anlysis V.
if Analysis V=blank then assign temp V:="***". end
else if VMin=blank and Analysis V not=blank then
assign global Out:="Y". assign temp V:=Analysis V.
end end end
if AlMax not=blank then
if Analysis Al not between AlMin to AlMax then
assign global Out:="Y". assign temp Al:=Analysis Al.
if Analysis Al=blank then assign temp Al:="***".
```

```
end end
else
if AlMin>Analysis Al then
assign global Out:="Y". assign temp Al:=Anlysis Al.
if Analysis Al=blank then assign temp Al:="***". end
else if AlMin=blank and Analysis Al not=blank then
assign global Out:="Y". assign temp Al:=Analysis Al.
end end end
if SbMax not=blank then
if Analysis Sb not between SbMin to SbMax then
assign global Out:="Y". assign temp Sb:=Analysis Sb.
if Analysis Sb=blank then assign temp Sb:="***".
end end
else
if SbMin>Analysis Sb then
assign global Out:="Y". assign temp Sb:=Anlysis Sb.
if Analysis Sb=blank then assign temp Sb:="***". end
else if SbMin=blank and Analysis Sb not=blank then
assign global Out:="Y". assign temp Sb:=Analysis Sb.
end end end
if SnMax not=blank then
if Analysis Sn not between SnMin to SnMax then
assign global Out:="Y". assign temp Sn:=Analysis Sn.
if Analysis Sn=blank then assign temp Sn:="***".
end end
else
if SnMin>Analysis Sn then
assign global Out:="Y". assign temp Sn:=Analysis Sn.
if Analysis Sn=blank then assign temp Sn:="***". end
else if SnMin=blank and Analysis Sn not=blank then
assign global Out:="Y". assign temp Sn:=Analysis Sn.
end end end
if CuMax not=blank then
if Analysis Cu not between CuMin to CuMax then
assign global Out:="Y". assign temp Cu:=Analysis Cu.
if Analysis Cu=blank then assign temp Cu:="***".
end end
else
if CuMin>Analysis Cu then
assign global Out:="Y". assign temp Cu:=Anlysis Cu.
if Analysis Cu=blank then assign temp Cu:="***". end
else if CuMin=blank and Analysis Cu not=blank then
assign global Out:="Y". assign temp Cu:=Analysis Cu.
end end end
if NMax not=blank then
if Analysis N not between NMin to NMax then
assign global Out:="Y". assign temp N:=Analysis N.
if Analysis N=blank then assign temp N:="***".
end end
else
if NMin>Analysis N then
assign global Out:="Y". assign temp N:=Anlysis N.
if Analysis N=blank then assign temp N:="***". end
else if NMin=blank and Analysis N not=blank then
assign global Out:="Y". assign temp N:=Analysis N.
end end end
```

```
if AsMax not=blank then
if Analysis As not between AsMin to AsMax then
assign global Out:="Y". assign temp As:=Analysis As.
if Analysis As=blank then assign temp As:="***".
end end
else
if AsMin>Analysis As then
assign global Out:="Y". assign temp As:=Analysis As.
if Analysis As=blank then assign temp As:="***". end
else if AsMin=blank and Analysis As not=blank then
assign global Out:="Y". assign temp As:=Analysis As.
end end end
if NbMax not=blank then
if Analysis Nb not between NbMin to NbMax then
assign global Out:="Y". assign temp Nb:=Analysis Nb.
if Analysis Nb=blank then assign temp Nb:="***".
end end
else
if NbMin>Analysis Nb then
assign global Out:="Y". assign temp Nb:=Anlysis Nb.
if Analysis Nb=blank then assign temp Nb:="***". end
else if NbMin=blank and Analysis Nb not=blank then
assign global Out:="Y". assign temp Nb:=Analysis Nb.
end end end
if TiMax not=blank then
if Analysis Ti not between TiMin to TiMax then
assign global Out:="Y". assign temp Ti:=Analysis Ti.
if Analysis Ti=blank then assign temp Ti:="***".
end end
else
if TiMin>Analysis Ti then
assign global Out:="Y". assign temp Ti:=Anlysis Ti.
if Analysis Ti=blank then assign temp Ti:="***". end
else if TiMin=blank and Analysis Ti not=blank then
assign global Out:="Y". assign temp Ti:=Analysis Ti.
end end end
if Ded1Max not=blank then
if Analysis D1Value not between Ded1Min to Ded1Max then
assign global Out:="Y". assign temp Ded1:=Analysis D1Value.
if Analysis D1Value=blank then assign temp Ded1:="***".
end end
else
if Ded1Min>Analysis D1Value then
assign global Out:="Y". assign temp Ded1:=Anlysis D1Value.
if Analysis D1Value=blank then assign temp Ded1:="***". end
else if Ded1Min=blank and Analysis D1Value not=blank then
assign global Out:="Y". assign temp Ded1:=Analysis D1Value.
end end end
if Ded2Max not=blank then
if Analysis D2Value not between Ded2Min to Ded2Max then
assign global Out:="Y". assign temp Ded2:=Analysis D2Value.
if Analysis D2Value=blank then assign temp Ded2:="***".
end end
else
if Ded2Min>Analysis D2Value then
assign global Out:="Y". assign temp Ded2:=Anlysis D2Value.
```

```
if Analysis D2Value=blank then assign temp Ded2:="***". end
else if Ded2Min=blank and Analysis D2Value not=blank then
assign global Out:="Y". assign temp Ded2:=Analysis D2Value.
end end end
if Analysis D3Value not between Ded3Min to Ded3Max then
assign global Out:="Y". assign temp Ded3:=Analysis D3Value.
if Analysis D3Value=blank then assign temp Ded3:="***".
end end
else
if Ded3Min>Analysis D3Value then
assign global Out:="Y". assign temp Ded3:=Anlysis D3Value.
if Analysis D3Value=blank then assign temp Ded3:="***". end
else if Ded3Min=blank and Analysis D3Value not=blank then
assign global Out:="Y". assign temp Ded3:=Analysis D3Value.
end end end
if Ded4Max not=blank then
if Analysis D4Value not between Ded4Min to Ded4Max then
assign global Out:="Y". assign temp Ded4:=Analysis D4Value.
if Analysis D4Value=blank then assign temp Ded4:="***".
end end
else
if Ded4Min>Analysis D4Value then
assign global Out:="Y". assign temp Ded4:=Anlysis D4Value.
if Analysis D4Value=blank then assign temp Ded4:="***". end
else if Ded4Min=blank and Analysis D4Value not=blank then
assign global Out:="Y". assign temp Ded4:=Analysis D4 Value.
end end end
if H2Max not=blank then
if Analysis H2 not between H2Min to H2Max then
assign global Out:="Y". assign temp H2:=Analysis H2.
if Analysis H2=blank then assign temp H2:="***".
end end
else
if H2Min>Analysis H2 then
assign global Out:="Y". assign temp H2:=Anlysis H2.
if Analysis H2=blank then assign temp H2:="***". end
else if H2Min=blank and Analysis H2 not=blank then
assign global Out:="Y". assign temp H2:=Analysis H2.
end end end
if global Out="Y" then
assign global R:=jointext(global R, global Out).
modify records in Analysis with(Cast=Analysis Cast and Cat=Analysis Cat)
RWS:=no.
List records
analysis cast;Analysis cat;
Issue_Steelord ST_No.;
Issue_Steelord Version; Ded1;Ded2;Ded3;Ded4;
CMin;SiMin;MnMin;PMin;SMin;CrMin;MoMin;NiMin;VMin;AlMin;SbMin;
SnMin;CuMin;NMin;AsMin;NbMin;TiMin;Ded1Min;Ded2Min;Ded3Min;
Ded4Min;H2Min;
CMax;SiMax;MnMax;PMax;SMax;CrMax;MoMax;NiMax;VMax;AlMax;SbMax;
SnMax;CuMax;NMax;AsMax;NbMax;TiMax;Ded1Max;Ded2Max;Ded3Max;
Ded4Max;H2Max;
temp C;temp Si;temp Mn;temp P;temp S;temp Cr;  temp Mo;temp Ni;temp V;
temp Al;temp Sb;  temp Sn;temp Cu;temp N;temp As;temp Nb;temp Ti;
temp Ded1;temp Ded2;temp Ded3;temp Ded4;temp H2.
```

```
end end end end
if global R=blank then
message "40,22,4,A11 the chemical analysis are|
within F.E.L. melting range." window.
end
break
End.
```

4-2-5-2 Forging Analysis Data-Entry

Procedure 1: *FSLladle*

Functions: This is a command procedure for controlling the process of data entry and checking of the FSL cast analysis. It runs two processing procedures below:-
Procedure 1 "Postdata3" is for posting CAST No. into Forging Status datafile for each FEL Order to create identity; Procedure 2 "Compare_Analysis" is for comparing cast analysis with FEL melting range specified in Steel Order Requirements and List any difference on the screen for users.

```
DQL QUERY
define global "CAST" Text 10. define global "Letter" Text 5.
define global "R" Text 3. define global "FEL Order No." Text 11.
define global "FEL Order 1" Numeric String 7. define global "FEL Order 2" Numeric String 7.
define global "FEL Order 3" Numeric String 7. define global "FEL Order 4" Numeric String 7.
define global "FEL Order No. 1" Numeric String 7. define global "Number" Number.
assign global FEL Order No. :=data-entry FEL Order No..
assign global FEL Order 1:=data-entry FEL Order 1.
assign global FEL Order 2:=data-entry FEL Order 2.
assign global FEL Order 3:=data-entry FEL Order 3.
assign global FEL Order 4:=data-entry FEL Order 4.
assign global CAST:=data-entry CAST.
assign global Letter:=data-entry Letter.
if data-entry Are you sure?=Yes then
run procedure "Postdata3".
message "Press SH-F3 Once Analysis File is Loaded.".
record entry "Analysis_Index".
if current status=2 or current status=3 then
run procedure "Compare_Analysis".
end
End.
```

Procedure 2: *Postdata3*
(Same as procedure 2 listed in section 4-2-4-2.)

Procedure 3: *Compare_Analysis*
(Same as procedure 2 listed in section 4-2-5-1.)

4-3 Procedures in Test Results Functions

4-3-1 Test Certificate Menu

4-3-1-1 Check Cast Analysis

Procedure 1: *Check_Analysis*
(Same as procedure 1 listed in section 4-2-5-1.)

4-3-1-2 Test Certificate Printing

Procedure 1 *Print_Cert.*

Functions: This is a command procedure for printing test certificate reports. It can check user's data entry to see if the report exists. If it is not, it will not call the processing procedure, Print_Test_Cert., therefore such error can be stopped at the early stage of the printing process.

Procedure 2: *Print_Test_Cert.*

Functions: This processing procedure is designed to print Test Certificate reports.

DQL QUERY
```
define global"Cert._No."Text 5. define global"Batch"Number. define global"Cert1_To"Text 15.
define global"Pos1"Text 2. define global"Pos2"Text 2. define global"Pos3"Text 2.
define global"Pos4"Text 2. define global"T_Ten1"Text 3. define global"T_Ten2"Text 3.
define global"T_Ten3"Text 3. define global"T_Ten4"Text 3. define global"Rp_Min1"Text 6.
define global"Rp_Min2"Text 6. define global"Rp_Min3"Text 6. define global"Rp_Min4"Text 6.
define global"Rp_Max1"Text 6. define global"Rp_Max2"Text 6. define global"Rp_Max3"Text 6.
define global"Rp_Max4"Text 6. define global"RM_Min1"Text 6. define global"RM_Min2"Text 6.
define global"RM_Min3"Text 6. define global"RM_Min4"Text 6. define global"RM_Max1"Text 6.
define global"RM_Max2"Text 6. define global"RM_Max3"Text 6. define global"RM_Max4"Text 6.
define global"A1"Text 3. define global"A2"Text 3. define global"A3"Text 3.
define global"A4"Text 3. define global"Z1"Text 3. define global"Z2"Text 3.
define global"Z3"Text 3. define global"Z4"Text 3. define global"T_Imp1"Text 4.
define global"T_Imp2"Text 4. define global"T_Imp3"Text 4. define global"T_Imp4"Text 4.
define global"Value1"Text 3. define global"Value2"Text 3. define global"Value3"Text 3.
define global"Value4"Text 3. define global"RV1"Text 3. define global"RV2"Text 3.
define global"RV3"Text 3. define global"RV4"Text 3. define global"HB_Min1"Text 4.
define global"HB_Min2"Text 4. define global"HB_Min3"Text 4. define global"HB_Min4"Text 4.
define global"HB_Max1"Text 4. define global"HB_Max2"Text 4. define global"HB_Max3"Text 4.
define global"HB_Max4"Text 4. define global"Shear1"Text 3. define global"Shear2"Text 3.
define global"Shear3"Text 3. define global"Shear4"Text 3. define global"Ang1"Text 3.
define global"Ang2"Text 3. define global"Ang3"Text 3. define global"Ang4"Text 3.
define global"Rad1"Text 3. define global"Rad2"Text 3. define global"Rad3"Text 3.
define global"Rad4"Text 3. define global"FATT1"Text 4. define global"FATT2"Text 4.
define global"FATT3"Text 4. define global"FATT4"Text 4. define global"LE1"Text 4.
define global"LE2"Text 4. define global"LE3"Text 4. define global"LE4"Text 4.
define global"Rem1"text 28. define global"Rem2"text 28. define global"Rem3"text 28.
define global"Rem4"text 28. define global"T_Code1"text 18. define global"T_Code2"text 18.
define global"Cat1"Text 8. define global"Cat2"Text 8. define global"Cat3"Text 8.
define global"Cat4"Text 8. define global"C_1"text 4. define global"C_2"text 4.
define global"C_3"text 4. define global"C_4"text 4. define global"Si_1"text 4.
define global"Si_2"text 4. define global"Si_3"text 4. define global"Si_4"text 4.
define global"Mn_1"text 5. define global"Mn_2"text 5. define global"Mn_3"text 5.
define global"Mn_4"text 5. define global"P_1"text 5. define global"P_2"text 5.
define global"P_3"text 5. define global"P_4"text 5. define global"S_1"text 5.
define global"S_2"text 5. define global"S_3"text 5. define global"S_4"text 5.
define global"Cr_1"text 5. define global"Cr_2"text 5. define global"Cr_3"text 5.
define global"Cr_4"text 5. define global"Mo_1"text 5. define global"Mo_2"text 5.
```

```
define global"Mo_3"text 5. define global"Mo_4"text 5. define global"Ni_1"text 5.
define global"Ni_2"text 5. define global"Ni_3"text 5. define global"Ni_4"text 5.
define global"V_1"text 4. define global"V_2"text 4. define global"V_3"text 4.
define global"V_4"text 4. define global"Al_1"text 5. define global"Al_2"text 5.
define global"Al_3"text 5. define global"Al_4"text 5. define global"Sb_1"text 6.
define global"Sb_2"text 6. define global"Sb_3"text 6. define global"Sb_4"text 6.
define global"Sn_1"text 5. define global"Sn_2"text 5. define global"Sn_3"text 5.
define global"Sn_4"text 5. define global"Cu_1"text 4. define global"Cu_2"text 4.
define global"Cu_3"text 4. define global"Cu_4"text 4. define global"N_1"text 5.
define global"N_2"text 5. define global"N_3"text 5. define global"N_4"text 5.
define global"As_1"text 5. define global"As_2"text 5. define global"As_3"text 5.
define global"As_4"text 5. define global"Ti_1"text 5. define global"Ti_2"text 5.
define global"Ti_3"text 5. define global"Ti_4"text 5. define global"Nb_1"text 5.
define global"Nb_2"text 5. define global"Nb_3"text 5. define global"Nb_4"text 5.
define global"H2_1"text 3. define global"H2_2"text 3. define global"H2_3"text 3.
define global"H2_4"text 3. define global"D1_1"text 5. define global"D1_2"text 5.
define global"D1_3"text 5. define global"D1_4"text 5. define global"D2_1"text 5.
define global"D2_2"text 5. define global"D2_3"text 5. define global"D2_4"text 5.
define global"D3_1"text 5. define global"D3_2"text 5. define global"D3_3"text 5.
define global"D3_4"text 5. define global"D4_1"text 5. define global"D4_2"text 5.
define global"D4_3"text 5. define global"D4_4"text 5.
for Certificate with Cert1_No.=global Cert._No. and Batch=global Batch;
list records
Cert1_No.; Batch; global Cert1_To; Test_Date;
Cast_No.; T_Unit; I_Unit; Stress; T_Name;
any Identification any SO_Status any Issued_Steelord
any FEL Range Quality;
any Identification any Customer Customer;
any Identification Description;
any Identification Customer Order No.;
any Identification Application;
any Identification Customer Drawing No.;
any Identification Drawing No.;
any Identification Inspection;
any Identification any Chem._Range Specification;
any Analysis with(Cast=Certificate Cast No.) Ded1;
any Analysis Ded2; any Analysis Ded3; any Analysis Ded4;
any Tests any Impact_Code Specimen_Size;
any Tests any Bend_Code Specimen_Size;
global Cat1;global Cat2;global Cat3;global Cat4;
global C_1;global C_2;global C_3;global C_4;
global Si_1;global Si_2;global Si_3;global Si_4;
global Mn_1;global Mn_2;global Mn_3;global Mn_4;
global P_1;global P_2;global P_3;global P_4;
global S_1;global S_2;global S_3;global S_4;
global Cr_1;global Cr_2;global Cr_3;global Cr_4;
global Mo_1;global Mo_2;global Mo_3;global Mo_4;
global Ni_1;global Ni_2;global Ni_3;global Ni_4;
global V_1;global V_2;global V_3;global V_4;
global Al_1;global Al_2;global Al_3;global Al_4;
global Sb_1;global Sb_2;global Sb_3;global Sb_4;
global Sn_1;global Sn_2;global Sn_3;global Sn_4;
global Cu_1;global Cu_2;global Cu_3;global Cu_4;
global N_1;global N_2;global N_3;global N_4;
global As_1;global As_2;global As_3;global As_4;
global Ti_1;global Ti_2;global Ti_3;global Ti_4;
```

```
global Nb_1;global Nb_2;global Nb_3;global Nb_4;
global D1_1;global D1_2;global D1_3;global D1_4;
global D2_1;global D2_2;global D2_3;global D2_4;
global D3_1;global D3_2;global D3_3;global D3_4;
global D4_1;global D4_2;global D4_3;global D4_4;
global H2_1;global H2_2;global H2_3;global H2_4;
global Pos1;global Pos2;global Pos3;global Pos4;
global T_Ten1;global T_Ten2; global T_Ten3;global T_Ten4;
global Rp_Min1;global Rp_Min2; global Rp_Min3;global Rp_Min4;
global Rp_Max1;global Rp_Max2; global Rp_Max3;global Rp_Max4;
global RM_Min1;global RM_Min2; global RM_Min3;global RM_Min4;
global RM_Max1;global RM_Max2; global RM_Max3;global RM_Max4;
global A1;globalA2;global A3;global A4;
global Z1;globalZ2;global Z3;global Z4;
global T_Imp1;global T_Imp2; global T_Imp3;global T_Imp4;
global Value1;global Value2; global Value3;global Value4;
global RV1;global RV2; global RV3;global RV4;
global HB_Min1;global HB_Min2;global HB_Min3;global HB_Min4;
global HB_Max1;global HB_Max2;global HB_Max3;global HB_Max4;
global Shear1;global Shear2;global Shear3;global Shear4;
global Ang1;global Ang2;global Ang3;global Ang4;
global Rad1;global Rad2;global Rad3;global Rad4;
global FATT1;global FATT2;global FATT3;global FATT4;
global LE1;global LE2;global LE3;global LE4;
global Rem1;global Rem2;global Rem3;global Rem4;
global T_Code1;global T_Code2;
 all Tests Test_No.; all Tests Pos.;
 all Tests Specimen1; all Tests TenTemp;
 all Tests Rp; all Tests Rm; all Tests A.; all Tests Z.;
 all Tests ImpTemp; all Tests Value1; all Tests Value2; all Tests Value3;
 all Tests HB/HV; all Tests Hardness;
 all Tests Shear1; all Tests Shear2; all Tests Shear3;
 all Tests Angle; all Tests Radius; all Tests FATT
 all Tests L.E1; all Tests L.E2; all Tests L.E3; all Tests Sentence;
 Long:other_test1; Long:other_test2; Long:other_test3;
 Long:other_test4; Long:other_test5; Long:condition1;
 Long:condition2; Long:condition3; Long:condition4;
 Results; Verdicts_by; Date.
End.
```

4-3-1-3 Test Certificate Design Data Entry

Procedure: *Test_Certificate*

Functions: This procedure is designed to control the process of data entry into test certificate data file. It can check user's data entry with the database records to avoid duplicated records being created.

4-3-1-4 View/Edit Test Certificate

Procedure: *View_Certificate*

Functions: This procedure is designed for user to view and edit test certificate records. Each time a modification is required, the procedure will check if that record has been approved and printed for customer or not. This is achieved by loading that record on a temporary file for user to view and edit.

4-3-2 NDT Report Menu

4-3-2-1 Start a NDT Report (first time)

Procedure 1: *NDT Report*

Functions: This is a command procedure which controls the process of view function on individual files. 1. View_U/S; 2. View_MPI; 3. View_Dye; 4. View_Visual; 5. View_Sulphur.

```
DQL QUERY
define global "Cast"text 10. define global "ID_No."numeric String 4.
define global "Ver."text 1. define global "FEL Order No."Numeric String 11.
define global "NDT_No."Text 5. define global "Op_No."Number.
define global "Batch"Number. define global "Exist"Text 3.
assign global FEL Order No.:=data-entry FEL Order No..
assign global Cast:=data-entry Cast. assign global ID_No.:=data-entry ID_No..
assign global Ver.:=data-entry Ver.. assign global NDT_No.:=data-entry NDT_No..
assign global Op_No.:=data-entry Op_No.. assign global Batch:=data-entry Batch.
assign global Exist:=data-entry Exist.
if data-entry are you sure?=Yes then
case (firstc(data-entry Cat's,3))
  value ("U/S"): run procedure "View_U/S".
  value ("Mag"): run procedure "View_MPI".
  value ("Dye"): run procedure "View_Dye".
  value ("S P"): run procedure "View_Sulphur".
  value ("Vis"): run procedure "View_Visual".
end
else
exit
End.
```

4-3-2-1 View/Edit NDT Reports

4-3-2-1-1 View/Edit Ultrasonic Report

Procedure 1: *View_U/S*

Functions: This is a processing procedure that can be run by a command procedure, NDT Report, or NDT Menu, View/Edit Ultrasonic Report. If it is run by the former, the global Exist should have already assigned a value of "New" or "Exist", which allows this procedure to enter basic information into U/S_Report datafile first, otherwise, it only allows user to access and edit the records in U/S_Report file.

```
DQL QUERY
define global "Cast" text 12. define global "ID_No." Numeric String 4.
define global "Ver." text 1. define global "NDT_No." text 5.
define global "Op_No." Number. define global "Batch" Number.
define global "Exist" Text 3. define global "FEL Order No." Numeric String 11.
if global Exist="New" then
enter a record in U/S_Report
NDT_No.:=global NDT_No.; Op_No.:=global Op_No.;
Batch:=global Batch; FEL Order No.:=global FEL Order No.;
Identity:=global Cast; ID_No.:=global ID_No.;
Ver.:=global Ver.; Date:=current date; Release:=no.
end
while current status not=1 do
if current status=2 then
```

```
message "Press SH-F3 once U/S_Report is loaded.".
else
message "Search using ALT-F1, SHFT-F3 or SHFT-F1.".
end
input using U/S_Report into "Temp_U/S".
if current status = 1 then
exit
end
for U/S_Report with NDT_No.=temp_U/S NDT_No. and Op_No.=Temp_U/S Op_No. and
                    Batch=Temp_U/S Batch;
if current status = 2 or current status = 3 then
if Release = no then
if Ready = no then
modify records
copy all from Temp_U/S.
else
message"{Note:} This is a report ready to print.|
All modified values are saved." window.
modify records
copy all from Temp_U/S.
end
for U/S_Scan with NDT_No.=temp_U/S NDT_No. and Op_No.=Temp_U/S Op_No. and
                    Batch=Temp_U/S Batch;
if direction=blank and Scan_No.=blank and Probe_No.=blank then
delete records
end end
else
message"The report has been printed for Cust.|
You are not allowed to modify it.|
Please contact with QA authority."window.
end
else
if current status = 4 and Ready = no and Release = no then
delete records in U/S_Scan with (NDT_No.=U/S_Report NDT_No. and Op_No.=
                    U/S_Report Op_No. and Batch=U/S_Report Batch;
delete records
end end end
End.
```

4-3-2-1-2 View/Edit M. Particle Report

4-3-2-1-3 View/Edit Dye Pen. Report

4-3-2-1-4 View/Edit Visual Report

4-3-2-1-5 View/Edit S. Print Report

(These procedures are similar to procedure, View/Edit U/S, listed in section 4-3-2-1-1

4-3-2-2 NDT Report Printing

4-3-2-2-1 Print U/S Test Report

Procedure 1: ***Print U/S Report***
Functions: This procedure is for printing Ultrasonic Test reports.

DQL QUERY
```
define temp "Atten1" Text 8. define temp "Atten2" Text 8.
define temp "Atten_dB1" Text 2. define temp "Atten_dB2" Text 3.
```

```
define temp "Coeff" Number.  define temp "Unit"Text 6.
if data-entry Unit:=mm then
assign temp Coeff := 1.  assign temp Unit:="mm".
else
assign temp Coeff := 25.2.  assign temp Unit:="inches".
end
if data-entry Sure = "no" then
exit
end
for U/S_Report  with NDT_No.=data-entry NDT No. and Op_No.= data-entry Op_No. and
                                Batch=data-entry Batch_No.;
if Ready = yes then
assign temp Atten1:=jointext(jointext(floor(Atten1/temp Coeff), "  "), temp Unit).
assign temp Atten2:=jointext(jointext(floor(Atten2/temp Coeff), "  "), temp Unit).
if Atten_dB1=blank then
assign temp Atten_dB1:="- -".
else assign temp Atten_dB1:=Atten_dB1.
end
 if Atten_dB2=blank then
assign temp Atten_dB2:="- -".
else assign temp Atten_dB2:=Atten_dB2.
end
if firstc(temp Atten1, 1)="0" then
assign temp Atten1:="- - - -".
end
if firstc(temp Atten2, 1)="0" then
assign temp Atten2:="- - - -".
end
if data-entry Release=yes then
if Release = no then
for U/S_Scan
enter a record in H_U/S_Scan
copy all from U/S_Scan.
end
end
modify records
Release:=data-entry Release.
end
List records
NDT_No.;  Op_No;  Batch;  FEL Order No.;
Quantity;  Identity;
any Identification any Customer Customer
any Identification Enquery No.;
any Identification Description;
any Identification Customer Order No.;
any Identification Customer Drawing No.;
any Identification Drawing No.;
Operator;  Qualification;
NDT_Spec.;  NDT_Spec.2;  NDT_Spec.3;
Date; Stage;  Surface;  Detector_#;
any NDT_Instrument Instrument_Type;
Couplant;  temp Atten_dB1;  temp Atten_dB2;
temp Atten1;  temp Atten2;
all U/S_Scan Direction in group;
all U/S_Scan Sacn_No. in order;
all U/S_Scan Probe_No.;
```

```
all U/S_Scan any Probe Wave;
all U/S_Scan any Probe Angle_Deg.;
all U/S_Scan any Probe Probe_Type;
all U/S_Scan any Probe FreqMHz;
all U/S_Scan any Probe Angle_Deg.;
all U/S_Scan any Probe Crystal_Size;
all U/S_Scan Text1;
all U/S_Scan Value1;
all U/S_Scan Text2;
all U/S_Scan Value2;
all U/S_Scan Plus;
all U/S_Scan Text3;
all U/S_Scan Value3;
Long:Note; Long:Note2; Long:Note3;
Results; Verdict; Evaluator; Sketch; Date2.
else
message"{Sorry}, This test record is not ready for printing."window.
end
End.
```

4-3-2-2-1 Print MPI Report
4-3-2-2-1 Print Dye Pen. Report
4-3-2-2-1 Print Visual Report
4-3-2-2-1 Print S. Print Report
(These procedures are similar to procedure, Print_U/S Report, listed in section 4-3-2-2-1

4-3-2-3 NDT Procedure Menu

4-3-2-3-1 NDE Procedure Data-Entry

Procedure 1: *Create_NDE*

Functions: This procedure is designed to control the process of data entry into the NDE procedure files. It can check user's request with the database file to see if the record has already existed or not. If it is a new record, the procedure will generate an identity number for it through its relationship with some files.

DQL QUERY
```
if data-entry Are you sure?=Yes then
 enter a record in NDE_Procedure
 copy all from data-entry;
 Data :=current data;
 Name :=current user name.
end.
```

4-3-2-3-2 Edit a NDE Procedure

Procedure 1: *Edit_NDE*

Functions: This procedure is designed for user to access and edit the NDE Procedures. It can also re-arrange the NDE procedure comments as specified by the user.

DQL QUERY
```
define temp "N" Number. define temp "LSN" Number. define temp "TN" Number.
while current status not=1 do
message"Search using ALT-F5, SHFT-F3 or SHFT-F1."
```

```
assign temp N:=blank.  assign temp LSN:=blank.
assign temp TN:=blank.
input using NDE_Procedure into "Temp_NDE".
if current status = 1 then
exit
end
if current status=2 or current status=3 then
message"Wait for DataEase processing please...".
for NDE_Procedure with(NDE_No.=Temp_NDE NDE_No. and Issue=Temp_NDE Issue);
modify records
copy all from Temp_NDE;  Date:=current date;
Name:=current user name.
for NDE_Tables
 if No.=blank then
 if Comment=blank then
 delete records
 end
 end
end
if current status=2 then
 for NDE_Tables
 assign temp TN:=temp TN+1.
 enter a record in RN_NDE
 copy all from NDE_Tables.
 delete records
 end
 assign temp TN:=temp TN+2.
 while temp N not=temp TN do
for RN_NDE with(No.=temp N and NDE_No.=Temp_NDE_No. and Issue=Temp_NDE Issue)
 assign temp LSN:=temp LSN+1.
 enter a record in NDE_Tables
 copy all from RN_NDE;
 No.:=temp LSN.
 delete records
end
assign temp N:=temp N+1.
end  end   end  end
End.
```

4-3-2-3-3 **Print a NDE Procedure**

Procedure 1: ***Print_NDE_Procedure***

Functions: This procedure is designed to print a specified NDE procedure report.

```
DQL QUERY
define global "NDE_No."text 8.  define global "Issue"text 1.
define temp "P_N" Number.  define temp "N" Number.  define temp "T_I"text.
assign global NDE_No.:=data-entry NDE_No..
assign global Issue:=data-entry Issue.
for NDE_Procedure with NDE_No.=global NDE_No. and Issue=global Issue;
 for NDE_Tables
 if  No. not=blank then
 assign temp N:=No..
 end
 assign temp TL:=temp TL+2.
```

```
 enter a record in RN_NDE
 copy all from NDE_Tables.
 end
while temp TL not=54 do
if temp TL=108 or temp TL=162 or temp TL=216 or temp TL=270 then
 break
 else
 assign temp TL:=temp TL+2.
 enter a record in RN_NDE
 NDE_No.:=global NDE_No.;  Issue:=global Issue;  No.:=temp N.
 end  end  end
for NDE_Procedure with NDE_No.=global NDE_No. and Issue=global Issue;
assign temp P_N:=ceil (temp TL/54).
list records
 NDE_No.; Issue;  Name;  Date;  temp P_N;
 all RN_NDE with(NDE_No.=global NDE_No. and Issue=global Issue) No.;
 all RN_NDE Comment.
 end
for RN_NDE with(NDE_No.=global NDE_No. and Issue=global Issue);
  delete records
End.
```

4-3-2-3-4 New Issue of NDE Procedure

Procedure 1: *Repeat_NDE*

Functions: This procedure is designed to assist user to create a new NDE procedure by repeating from previous NDE procedures.

DQL QUERY

```
define global "NDE_I"Text 1.  define global "NDE_No."Text 5.  define global "Issue"Text 1.
if data-entry Are you sure?=Yes then
 enter a record in NDE Procedure
 NDE_No.:=data-entry NDE_No.;  Issue:=data-entry Issue;
 Name:=current user name;  Date:=current date.
  for NDE_Table with NDE_No.=data-entry P_NDE_No. and Issue=data-entry P_Issue;
 enter a record in RN_NDE
 copy all from NDE_Tables;
 NDE_No.:=data-entry NDE_No.;  Issue:=data-entry Issue.
 end
 for RN_NDE with NDE_No.=data-entry NDE_No. and Issue=data-entry issue;
 enter a record in NDE_Tables
 copy all from RN_NDE.
 end
 message"Record created, please edit it."window.
 else
 message"Not being confirmed, you should press ESC, exit now."window.
End.
```

APPENDIX 5

LISTING OF EXERT SYSTEM RULES

```
* RULE LIST                Wed Nov 30 09:38:34 1993

*
[  1] apply the answers to the questions            Sp
      IF        DO: Assign Variable
                score:=20
    +  AND [  51] T-percentages being processed
    +  AND [  32] Initialisation
      AND       DO: Assign Variable
                score:=40
    +  AND [  51] T-percentages being processed
    +  AND [   8] assign parameters
      AND       DO: Assign Variable
                score:=60
    +  AND [  51] T-percentages being processed
    +  AND [  18] compare the parameters

[  2] ask all the questions                  Sp
    + IF   [  10] check ingot type ( >="Igt_X" )
    +  AND [  11] check Ingot Weights
    +  AND [  12] check liquid WT.
    +  AND [  16] check spare metal
    +  AND [  13] check number of ladles
    +  AND [  17] check VCD
    +  AND [  48] Steel Grade or Al treated
    +  AND [  15] check over heat temperature
    +  AND [  14] check other parameters
    +  AND [  63] what processes are supposed

[  3] ask for head filled to                 Sp
    F IF        DO: Menu Question  Head$

                       CRYSTAL EXPERT SYSTEM
                 STEELMAKING PROCEDURE APPROVAL EXPERT


                   WHERE IS THE INGOT HEAD FILLED TO?

                             {3/4  }
                             {Notch} minimum

             Select answer by moving the cursor up and down keys

                         Press Enter when ready
        COL : SURR     White on Blue
        COL : 0,0      Blue on Cyan       COL : 4,0      White on Black
        COL : 5,0      White on Blue      COL : 19,0     Lt_Gray on Black
        COL : 20,0     Blue on Cyan
        MENU: 11,32,38  Blue on Gray          Lt_Red on Blue
        MENU: 12,32,38  Blue on Gray          Lt_Red on Blue
```

OR DO: Succeed

[4] ask for new spray limiter Sp
F IF DO: Menu Question spray$

CRYSTAL EXPERT SYSTEM
STEELMAKING PROCEDURE APPROVAL EXPERT

IS THE SPRAY LIMITER APPLIED?

{YES} OR {NO}

Select answer by moving the cursor right and left keys

Press Enter when ready

COL : SURR	White on Blue		
COL : 0,0	Blue on Cyan	COL : 4,0	White on Black
COL : 5,0	White on Blue	COL : 19,0	Lt_Gray on Black
COL : 20,0	Blue on Cyan		
MENU: 11,32,36	Blue on Gray		Lt_Red on Blue
MENU: 11,41,44	Blue on Gray		Lt_Red on Blue

OR DO: Succeed

[5] ask for nozzle size Sp
F IF DO: Menu Question NOZZLE$

CRYSTAL EXPERT SYSTEM
STEELMAKING PROCEDURE APPROVAL EXPERT

WHAT IS THE NOZZLE SIZE?

{X_{1-1} mm}
{X_{1-2} mm}
{X_{1-3} mm}
{X_{1-4} mm}
{Others }

Select answer by moving the cursor up and down keys

Press Enter when ready

COL : SURR	White on Blue		
COL : 0,0	Blue on Cyan	COL : 4,0	White on Black
COL : 5,0	White on Blue	COL : 19,0	Lt_Gray on Black
COL : 20,0	Blue on Cyan		
MENU: 10,35,41	Blue on Gray		Lt_Red on Blue

```
        MENU: 11,35,41  Blue on Gray        Lt_Red on Blue
        MENU: 12,35,41  Blue on Gray        Lt_Red on Blue
        MENU: 13,35,41  Blue on Gray        Lt_Red on Blue
        MENU: 14,35,41  Blue on Gray        Lt_Red on Blue

   AND      DO: Test Expression
            NOZZLE$<>"other"
 + AND [  6] ask for the teeming rate

 F OR       DO: Display Form

                       CRYSTAL EXPERT SYSTEM
              STEELMAKING PROCEDURE APPROVAL EXPERT


                      SPECIFY THE NOZZLE SIZE?


                             <NOZ> mm


        Select answer by moving the cursor up and down keys

                        Press Enter when ready
        COL : SURR     White on Blue
        COL : 0,0      Blue on Cyan      COL : 4,0     White on Black
        COL : 5,0      White on Blue     COL : 19,0    Lt_Gray on Black
        COL : 20,0     Blue on Cyan
        IN  : 11,35,39,0 Blue on Gray
        NOZZLE

   AND      DO: Assign Variable
          NOZZLE$:=string$(NOZZLE,3,0)
 + AND [  6] ask for the teeming rate

[  6] ask for the teeming rate               Sp
   F IF       DO: Menu Question  rate$

                       CRYSTAL EXPERT SYSTEM
              STEELMAKING PROCEDURE APPROVAL EXPERT


                   For the nozzle size: [NOZZLE$]

                      what is the teeming rate?


                            {Full     }
                            {Throttled}

        Select answer by moving the cursor up and down keys
                        Press Enter when ready
        COL : SURR     White on Blue
        COL : 0,0      Blue on Cyan      COL : 4,0     White on Black
        COL : 5,0      White on Blue     COL : 19,0    Lt_Gray on Black
        COL : 20,0     Blue on Cyan
```

```
            MENU: 13,33,43   Blue on Gray          Lt_Red on Blue
            MENU: 14,33,43   Blue on Gray          Lt_Red on Blue
            OUT : 7,42,50
               NOZZLE$

      OR         DO: Succeed

[  7] assign a bar
      IF        DO: Assign Variable
                k:=k+1
       AND        DO: Test Expression
                k<=25
       AND        DO: Assign Variable
                bar$:=bar$+chr$(219)
       AND        DO: Restart Rule

      OR         DO: Succeed

[  8] assign parameters                          Sp
     + IF   [ 41] parameters for ingot >="Igt_X"
     +  AND [ 52] teeming temperature over Liquid temperature
     +  AND [ 49] steelmaking processes

[  9] assign processes                           Sp
      IF        DO: Assign Variable
                Process1$:=P$[P1]
       AND        DO: Assign Variable
                Process2$:=P$[P2]

[ 10] check ingot type ( >="Igt_X" )                    Sp
     F IF       DO: Display Form

                        CRYSTAL EXPERT SYSTEM
                 STEELMAKING PROCEDURE APPROVAL EXPERT


                     ÉÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ»
                     º WHAT IS THE INGOT SIZE?  º
                     º                          º
                     º                          º
                     º        <IN> inches       º
                     ÈÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ¼


                Enter the ingot size intothe field directly

                       Press Enter when ready
           COL : SURR     White on Blue
           COL : 0,0     Blue on Cyan       COL : 4,0      White on Black
           COL : 5,0     White on Blue      COL : 19,0     Lt_Gray on Black
           COL : 20,0     Blue on Cyan
           IN  : 11,35,38,0  Blue on Gray
              INGOT

       AND        DO: Test Expression
               INGOT>=Igt_X
```

```
 OR        DO: Test Expression
           INGOT<>0&INGOT<Igt_X
   AND       DO: Test Expression
           beep(824,3)
F  AND       DO: Display Form

                 CRYSTAL EXPERT SYSTEM
            STEELMAKING PROCEDURE APPROVAL EXPERT

            ÉÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ»
            º WHAT IS THE INGOT SIZE?  º
            º                          º
            º    [IN]  inches          º
            ÉÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ»

            ÈÍÍÍÍËÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÊÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ»
            º Sorry! We do not have any solutions for which º
            º       the ingots are not over or equal to     º
            º              a specific diameter.             º
            º                                               º
            º       Press any key to re-enter it again.     º
            ÈÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ¼
     COL : SURR    White on Blue
     COL : 0,0     Blue on Cyan     COL : 4,0    White on Black
     COL : 5,0     White on Blue    COL : 19,0   Lt_Gray on Black
     COL : 20,0    Blue on Cyan
     OUT : 11,33,36,0
        INGOT

+  AND       DO: Wipe Rule
           ask all the questions
   AND       DO: Restart Rule
 OR        DO: Test Expression
           INGOT=0
   AND       DO: Test Expression
           beep(824,3)
F  AND       DO: Display Form

                 CRYSTAL EXPERT SYSTEM
            STEELMAKING PROCEDURE APPROVAL EXPERT

            ÉÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ»
            º       WHAT IS THE INGOT SIZE?       º
            º                                     º
            º                                     º
            º              [IN]  inches           º
            ÈÍÍÍÍÍÍÍËÍÍÍÍÍÍÍÍÍÍÍÍÍÍÊÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ»
            º Sorry! The expert can not find solutions if º
            º       you don't specify the ingot size.     º
            º                                             º
            º             Press any key to exit!          º
            ÈÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ¼
```

Enter the ingot size intothe field directly

Press Enter when ready

COL : SURR White on Blue
COL : 0,0 Blue on Cyan COL : 4,0 White on Black
COL : 5,0 White on Blue COL : 19,0 Lt_Gray on Black
COL : 20,0 Blue on Cyan
OUT : 11,33,36,0
INGOT

AND DO: Fail

[11] check Ingot Weights Sp
F IF DO: Display Form

CRYSTAL EXPERT SYSTEM
STEELMAKING PROCEDURE APPROVAL EXPERT

PLEASE SPECIFY THE INGOT WEIGHT:

<Wt > TONNES

Enter the ingot weight value directly into the field

Press Enter when ready

COL : SURR White on Blue
COL : 0,0 Blue on Cyan COL : 4,0 White on Black
COL : 5,0 White on Blue COL : 19,0 Lt_Gray on Black
COL : 20,0 Blue on Cyan
IN : 12,31,39,0 Blue on Gray
Wt

[12] check liquid WT. Sp
F IF DO: Display Form

CRYSTAL EXPERT SYSTEM
STEELMAKING PROCEDURE APPROVAL EXPERT

PLEASE SPECIFY THE APPROX LIQUID WT.:

<liquid > TONNES

Enter the liquid weight into the field directly

Press Enter when ready

COL : SURR White on Blue
COL : 0,0 Blue on Cyan COL : 4,0 White on Black
COL : 5,0 White on Blue COL : 19,0 Lt_Gray on Black
COL : 20,0 Blue on Cyan
IN : 11,32,42,0 Blue on Gray
liquid

OR DO: Succeed

[13] check number of ladles Sp
IF DO: Succeed
F AND DO: Menu Question ladles$

CRYSTAL EXPERT SYSTEM
STEELMAKING PROCEDURE APPROVAL EXPERT

PLEASE SPECIFY THE NUMBER OF LADLES:

1. {ONE LADLE}
2. {TWO LADLES}
3. {THREE LADLES}

Select answer by moving the cursor up and down keys

Press Enter when ready

COL : SURR White on Blue
COL : 0,0 Blue on Cyan COL : 4,0 White on Black
COL : 5,0 White on Blue COL : 19,0 Lt_Gray on Black
COL : 20,0 Blue on Cyan
MENU: 11,32,42 Blue on Gray Lt_Red on Blue
MENU: 12,32,43 Blue on Gray Lt_Red on Blue
MENU: 13,32,45 Blue on Gray Lt_Red on Blue

[14] check other parameters Sp
+ IF [4] ask for new spray limiter
+ AND [5] ask for nozzle size
+ AND [3] ask for head filled to

[15] check over heat temperature Sp
F IF DO: Display Form

CRYSTAL EXPERT SYSTEM
STEELMAKING PROCEDURE APPROVAL EXPERT

PLEASE SPECIFY THE TEMPERATURE ABOVE LIQUIDUS:

LIQUIDUS + <Tem> C

Enter the temperature directly into the field

Press Enter when ready

COL : SURR White on Blue
COL : 0,0 Blue on Cyan COL : 4,0 White on Black
COL : 5,0 White on Blue COL : 19,0 Lt_Gray on Black
COL : 20,0 Blue on Cyan
IN : 12,40,44,0 Blue on Gray
Temp

[16] check spare metal Sp
F IF DO: Display Form

CRYSTAL EXPERT SYSTEM
STEELMAKING PROCEDURE APPROVAL EXPERT

PLEASE SPECIFY THE SPARE METAL:

<metal > TONNES

Enter the spare metal weight directly into the field

Press Enter when ready

COL : SURR White on Blue
COL : 0,0 Blue on Cyan COL : 4,0 White on Black
COL : 5,0 White on Blue COL : 19,0 Lt_Gray on Black
COL : 20,0 Blue on Cyan
IN : 12,31,39,0 Blue on Gray
metal

[17] check VCD Sp
F IF DO: Menu Question VCD$

CRYSTAL EXPERT SYSTEM
STEELMAKING PROCEDURE APPROVAL EXPERT

IS THE STEEL VCD TREATED?

{YES} OR {NO}

Select answer by moving the cursor right and left keys

Press Enter when ready

COL : SURR White on Blue
COL : 0,0 Blue on Cyan COL : 4,0 White on Black
COL : 5,0 White on Blue COL : 19,0 Lt_Gray on Black
COL : 20,0 Blue on Cyan
MENU: 11,32,36 Blue on Gray Lt_Red on Blue
MENU: 11,41,44 Blue on Gray Lt_Red on Blue

OR DO: Succeed

[18] compare the parameters Sp
+ IF [28] ingot weight
+ AND [40] over heat Temp.
+ AND [37] new pray limiter
+ AND [39] nozzle size
+ AND [27] head filled to
+ AND [45] processes

```
[ 19] confirm to exit                         Sp
      IF        DO: Test Expression
                exit$="Yes"
        AND       DO: Quit

      OR        DO: Succeed

[ 20] determine the next step...
     + IF   [ 60] user selects from the "what-next" menu
        AND       DO: Test Expression
                (next=1)|(next=4)
     +  AND       DO: Wipe Rule
                user selects from the "what-next" menu

      OR        DO: Test Expression
                next=2
     +  AND       DO: Wipe Rule
                user selects from the "what-next" menu
        AND       DO: Restart Rule

      OR        DO: Test Expression
                next=3
        AND       DO: Assign Variable
                next:=2
        AND       DO: Restart Rule


[ 21] display results                         Sp
     + IF   [ 26] find "not" or "hasn't"
        AND       DO: Assign Variable
                score:=100
     +  AND [ 51] T-percentages being processed
     +  AND [ 22] displaying result 1
     +  AND [ 46] return to the main menu

     + OR   [ 23] displaying result 2
     +  AND [ 46] return to the main menu

[ 22] displaying result 1                     Sp
     + IF   [ 35] menu 1
     +  AND [ 55] test if print selected 1

      OR        DO: Succeed

[ 23] displaying result 2                     Sp
     + IF   [ 36] menu 2
     +  AND [ 56] test if print selected 2

      OR        DO: Succeed

[ 24] find "hasn't"                           Sp
      IF        DO: Assign Variable
                J:=J+1
        AND       DO: Test Expression
                J<arrsize(arrR1$[#],0)
        AND       DO: Assign Variable
```

```
            F$:=query$(find("hasn't",arrR1$[J]),F$+"",F$+"Y")
      AND      DO: Restart Rule
      OR       DO: Test Expression   start("Y",F$)
      AND      DO: Succeed

[ 25] find "not"                        Sp
    IF        DO: Assign Variable
              J:=J+1
     AND        DO: Test Expression
              J<arrsize(arrR1$[#],0)
     AND        DO: Assign Variable
              F$:=query$(find("not",arrR1$[J]),F$+"",F$+"Y")
     AND        DO: Restart Rule

    OR        DO: Test Expression   start("Y",F$)
     AND        DO: Succeed

[ 26] find "not" or "hasn't"                 Sp
    IF        DO: Assign Variable
              J:=0
  +  AND [ 25] find "not"

    OR        DO: Assign Variable
              J:=0
  +  AND [ 24] find "hasn't"

[ 27] head filled to                    Sp
    IF        DO: Assign Variable
              J:=J+1
     AND        DO: Test Expression
              Head$=E_Head$
     AND        DO: Assign Variable
              arrR1$[J]:="The ingot head filled to is OK."

    OR        DO: Assign Variable
              arrR1$[J]:="The ingot head filled to is not OK."
     AND        DO: Succeed

[ 28] ingot weight                      Sp
    IF        DO: Assign Variable
              J:=J+1
     AND        DO: Test Expression
              liquid-Wt<=metal|metal<X3
     AND        DO: Assign Variable
              arrR1$[J]:="The spare metal is not sufficient."

    OR        DO: Test Expression
              metal>=X3
     AND        DO: Assign Variable
              arrR1$[J]:="The metal is sufficient."

[ 29] Init
    IF        DO: Test Expression
              arrclr(arr1$[#])
     AND        DO: Assign Variable
              l1$:="      CRYSTAL EXPERT SYSTEM"
```

```
    AND     DO: Assign Variable
            l2$:="STEELMAKING PROCEDURE APPROVAL EXPERT"
    AND     DO: Assign Variable
            I:=0

[ 30] initail variables
   IF       DO: Assign Variable
            INGOT:=0
    AND     DO: Assign Variable
            F$:=""

[ 31] initial screen
   + IF   [ 62] Welcome screen
     AND    DO: Fail

   + OR   [ 29] Init
   +  AND [ 47] set line to next spot in array
   F  AND    DO: View Form
                        [arr1$[1]              ]
                        [arr1$[2]              ]
                        [arr1$[3]              ]
                        [arr1$[4]              ]
                        [arr1$[5]              ]
                        [arr1$[6]              ]
                        [arr1$[7]              ]
                        [arr1$[8]              ]
                        [arr1$[9]              ]
                        [arr1$[10]             ]
                        [arr1$[11]             ]
            COL : SURR     White on Blue
            COL : 0,0    Blue on Cyan      COL : 4,0     White on Black
            COL : 4,21   Lt_Cyan on Black    COL : 4,63    White on Black
            COL : 5,0    White on Blue     COL : 19,0    Lt_Gray on Black
            COL : 20,0   Black on Cyan
            OUT : 1,21,61
               arr1$[1]
            OUT : 2,21,61
               arr1$[2]
            OUT : 3,21,61
               arr1$[3]
            OUT : 4,21,61
               arr1$[4]
            OUT : 5,21,61
               arr1$[5]
            OUT : 6,21,61
               arr1$[6]
            OUT : 7,21,61
               arr1$[7]
            OUT : 8,21,61
               arr1$[8]
            OUT : 9,21,61
               arr1$[9]
            OUT : 10,21,61
               arr1$[10]
            OUT : 11,21,61
               arr1$[11]
```

```
     AND       DO: Restart Rule

[ 32] Initialisation                    Sp
     IF        DO: Test Expression
               arrclr(arrR1$[#])
     AND       DO: Assign Variable
               J:=0

[ 33] Introduction                      Sp
   + IF    [ 31] NOT initial screen

     OR        DO: Succeed

[ 34] Main Menu
   F IF        DO: Menu Question  mode

                    CRYSTAL EXPERT SYSTEM
             STEELMAKING PROCEDURE APPROVAL EXPERT

          Éíííííííííííííííííííííííííííííííííííííííííííííííííííí»
          º                                               º
          º {Approve FSL Steelmaking Procedures}          º
          º {Consult Expert for Information}              º
          º {Quit to DOS                 }                º
          º                                               º
          Èíííííííííííííííííííííííííííííííííííííííííííííííííííí¼


          Select answer by moving the cursor up and down keys

                      Press Enter when ready
        COL : SURR     White on Blue
        COL : 0,0      Blue on Cyan      COL : 4,0     White on Black
        COL : 5,0      White on Blue     COL : 19,0    Lt_Gray on Black
        COL : 20,0     Blue on Cyan
        MENU: 8,23,54  Blue on Gray        Lt_Red on Blue
        MENU: 9,23,54  Blue on Cyan        Red on Cyan
        MENU: 10,23,54 Blue on Cyan        Red on Cyan

     AND       DO: Test Expression
               mode=1

     OR        DO: Test Expression
               mode=2
   F AND       DO: Menu Question  mode1

                    CRYSTAL EXPERT SYSTEM
             STEELMAKING PROCEDURE APPROVAL EXPERT

          Are you sure you want to consult with the Expert system?


          Select answer by moving the cursor up and down keys

                      Press Enter when ready
```

```
            COL : SURR      White on Blue
            COL : 0,0       Blue on Cyan        COL : 4,0       White on Blue
            COL : 4,35      White on Black      COL : 5,0       White on Blue
            COL : 19,0      Lt_Gray on Black    COL : 20,0      Blue on Cyan

   AND       DO: Fail

  OR        DO: Test Expression
            mode=3
   AND       DO: Test Expression
            beep(824,3)
F  AND       DO: Menu Question  exit$

                         CRYSTAL EXPERT SYSTEM
                  STEELMAKING PROCEDURE APPROVAL EXPERT


                  Éííííííííííííííííííííííííííííííííííííííííííííííííí»
                  º                                                º
                  º Approve FSL Steelmaking Procedures             º
                  º Consult Expert for Information                 º
                  º Quit to DOS                                    º
                  º                                                º
                  ÈííííííííííííííÉíííííííííííííííííííííííííííííííííí»
                  º                                       º
                  º DO YOU REALLY WANT TO EXIT?           º
                  º                                       º
                  º     {YES} OR {NO}                     º
                  º                                       º
                  Èíííííííííííííííííííííííííííííííííííííííí¼

            Select answer by moving the cursor right and left keys
                          Press Enter when ready
            COL : SURR      White on Blue
            COL : 0,0       Blue on Cyan        COL : 4,0       White on Black
            COL : 5,0       White on Blue       COL : 19,0      Lt_Gray on Black
            COL : 19,20     White on Blue       COL : 19,77     Lt_Gray on Black
            COL : 20,0      Blue on Cyan
            MENU: 16,48,52  Blue on Gray            Lt_Red on Blue
            MENU: 16,57,60  Blue on Gray            Lt_Red on Blue

 +  AND [  19] confirm to exit
    AND       DO: Assign Variable
            mode:=0
    AND       DO: Restart Rule

[ 35] menu 1                         Sp
    F IF      DO: Menu Question  menuvar1

                         CRYSTAL EXPERT SYSTEM
                  STEELMAKING PROCEDURE APPROVAL EXPERT

               Éííííííííííííííííííííííííííííííííííííííííííííííííííííííí»
               º The Steelmaking Procedure is not accepatable as       º
               º   the Expert System has evaluated below,              º
```

```
°                                                    °
° [arrR1$[1]                  ]                      °
° [arrR1$[2]                  ]                      °
° [arrR1$[3]                  ]                      °
° [arrR1$[4]                  ]                      °
° [arrR1$[5]                  ]                      °
° [arrR1$[6]                  ]                      °
° [arrR1$[7]                  ]                      °
Èííííííííííííííííííííííííííííííííííííííííííííííííííííí¼
° The action to be taken: {Restart} or {Print}       °
Èííííííííííííííííííííííííííííííííííííííííííííííííííííí¼
  Select answer by moving the cursor up and down keys
                 Press Enter when ready
COL : SURR     White on Blue
COL : 0,0      Blue on Cyan        COL : 4,0     White on Blue
COL : 4,35     White on Black      COL : 5,0     White on Blue
COL : 5,26     White on Black      COL : 5,28    White on Blue
COL : 19,0     Lt_Gray on Black    COL : 19,35   White on Blue
COL : 19,40    Lt_Gray on Black    COL : 20,0    Blue on Cyan
MENU: 18,41,49  Blue on Gray           Lt_Red on Blue
MENU: 18,54,60  Blue on Gray           Lt_Red on Blue
OUT : 9,16,61
   arrR1$[1]
OUT : 10,16,61
   arrR1$[2]
OUT : 11,16,61
   arrR1$[3]
OUT : 12,16,61
   arrR1$[4]
OUT : 13,16,61
   arrR1$[5]
OUT : 14,16,61
   arrR1$[6]
OUT : 15,16,61
   arrR1$[7]

[ 36] menu 2                        Sp
      F IF      DO: Menu Question  menuvar2

                 CRYSTAL EXPERT SYSTEM
         STEELMAKING PROCEDURE APPROVAL EXPERT

Éííííííííííííííííííííííííííííííííííííííííííííííííííííí»
° The Steelmaking Procedure has beeen approved by    °
°     the Expert System as shown below,              °
°                                                    °
° [arrR1$[1]                  ]                      °
° [arrR1$[2]                  ]                      °
° [arrR1$[3]                  ]                      °
° [arrR1$[4]                  ]                      °
° [arrR1$[5]                  ]                      °
° [arrR1$[6]                  ]                      °
° [arrR1$[7]                  ]                      °
Èííííííííííííííííííííííííííííííííííííííííííííííííííííí¼
°  The action to be taken: {Restart} or {Print}      °
Èííííííííííííííííííííííííííííííííííííííííííííííííííííí¼
```

Select answer by moving the cursor up and down keys

Press Enter when ready

```
COL : SURR      White on Blue
COL : 0,0       Blue on Cyan        COL : 4,0     White on Blue
COL : 4,35      White on Black      COL : 5,0     White on Blue
COL : 5,26      White on Black      COL : 5,28    White on Blue
COL : 19,0      Lt_Gray on Black    COL : 20,0    Blue on Cyan
MENU: 18,41,49  Blue on Gray        Lt_Red on Blue
MENU: 18,54,60  Blue on Gray        Lt_Red on Blue
OUT : 9,16,61
   arrR1$[1]
OUT : 10,16,61
   arrR1$[2]
OUT : 11,16,61
   arrR1$[3]
OUT : 12,16,61
   arrR1$[4]
OUT : 13,16,61
   arrR1$[5]
OUT : 14,16,61
   arrR1$[6]
OUT : 15,16,61
   arrR1$[7]

[ 37] new pray limiter                   Sp
      IF     DO: Assign Variable
             J:=J+1
       AND     DO: Test Expression
             spray$="Yes"
       AND     DO: Assign Variable
             arrR1$[J]:="The spray limiter has been applied."

      OR      DO: Assign Variable
             arrR1$[J]:="The spray limiter hasn't been applied"
       AND     DO: Succeed

[ 38] not VCD treated steel
      IF     DO: Test Expression
             VCD$="No"

[ 39] nozzle size                        Sp
      IF     DO: Assign Variable
             J:=J+1
       AND     DO: Test Expression
             NOZZLE$<>E_Nozzle$
       AND     DO: Test Expression
             NOZZLE$<"X1 mm"
       AND     DO: Assign Variable
             arrR1$[J]:="The nozzle size is not proper."

      OR      DO: Test Expression
             NOZZLE$="X1 mm"
       AND     DO: Assign Variable
             arrR1$[J]:="The nozzle size is OK."
```

```
OR        DO: Test Expression
          NOZZLE$>"X1 mm"
  AND       DO: Test Expression
          rate$="Throttled"
  AND       DO: Assign Variable
          arrR1$[J]:="The nozzle size is OK."
  AND       DO: Succeed

[ 40] over heat Temp.                   Sp
    IF      DO: Assign Variable
            J:=J+1
     AND      DO: Test Expression
            Temp<E_Temp
     AND      DO: Assign Variable
            arrR1$[J]:="The over heat Temp. is not sufficient"

    OR       DO: Assign Variable
            arrR1$[J]:="The over heat Temp. is OK."
     AND      DO: Succeed

[ 41] parameters for ingot >="Igt_X"          Sp
    IF      DO: Assign Variable
            E_metal:=X3
     AND      DO: Assign Variable
            E_Nozzle$:="X1 mm"
     AND      DO: Assign Variable
            E_spray$:="Yes"
     AND      DO: Assign Variable
            E_Head$:="Notch"

[ 42] process for ingot weight between X3a - X3b tonnes
    IF      DO: Test Expression
            Wt>=X3a
     AND      DO: Test Expression
            Wt<=X3b

[ 43] process for ingot weight between X1a - X1b tonnes
    IF      DO: Test Expression
            Wt>=X1a
     AND      DO: Test Expression
            Wt<=X1b

[ 44] process for ingot weight betweenX2a - X2b tonnes
    IF      DO: Test Expression
            Wt>=X2a
     AND      DO: Test Expression
            Wt<=X2b

[ 45] processes                     Sp
    IF      DO: Assign Variable
            J:=J+1
     AND      DO: Test Expression
            Process1$=E_Process1$|Process1$=E_Process2$
     AND      DO: Test Expression
            Process2$=E_Process2$|Process2$=E_Process1$
```

```
  AND       DO: Assign Variable
            arrR1$[J]:="The processes are OK."

  OR        DO: Assign Variable
            arrR1$[J]:="The processes are not proper."
  AND       DO: Succeed

[ 46] return to the main menu                    Sp
  IF        DO: Assign Variable
            score:=0
+ AND [ 34] Main Menu

[ 47] set line to next spot in array             Sp
  IF        DO: Test Expression
            I<10
  AND       DO: Test Expression
            arrclr(arr1$[#])
  AND       DO: Assign Variable
            arr1$[10-I]:=l1$
  AND       DO: Assign Variable
            arr1$[11-I]:=l2$
  AND       DO: Assign Variable
            I:=I+1

[ 48] Steel Grade or Al treated                  Sp
  F IF      DO: Menu Question  Steel$

                    CRYSTAL EXPERT SYSTEM
              STEELMAKING PROCEDURE APPROVAL EXPERT

                IS THE STEEL 1CrMoV or Al TREATED?


                        {YES} OR {NO}


          Select answer by moving the cursor right and left keys

                       Press Enter when ready
        COL : SURR      White on Blue
        COL : 0,0       Blue on Cyan      COL : 4,0     White on Black
        COL : 5,0       White on Blue     COL : 19,0    Lt_Gray on Black
        COL : 20,0      Blue on Cyan
        MENU: 12,31,35  Blue on Gray            Lt_Red on Blue
        MENU: 12,40,43  Blue on Gray            Lt_Red on Blue

  OR        DO: Display Form
            Sorry! The expert system can not find any solution
               if you don't spscify the Steel grade or the
               deoxidation parameter.


               Press any key to exit!
        COL : SURR      White on Blue
        COL : 0,0       White on Blue
```

```
        AND      DO: Fail

[ 49] steelmaking processes                    Sp
      + IF   [ 61] VCD treated steel
      +  AND [ 43] process for ingot weight between X1a - X1btonnes
        AND      DO: Assign Variable
                 E_Process1$:="Single Slag Furnace"
        AND      DO: Assign Variable
                 E_Process2$:="Single Slag VAD degas"

      + OR   [ 61] VCD treated steel
      +  AND [ 44] process for ingot weight between X2a - X2b tonnes
        AND      DO: Assign Variable
                 E_Process1$:="Single Slag Furnace"
        AND      DO: Assign Variable
                 E_Process2$:="2 Single Slag VAD/LF degas"

      + OR   [ 38] not VCD treated steel
      +  AND [ 43] process for ingot weight between X1a - X1b tonnes
        AND      DO: Assign Variable
                 E_Process1$:="Double Slag Furnace"
        AND      DO: Assign Variable
                 E_Process2$:="VAD Not Specified"

      + OR   [ 38] not VCD treated steel
      +  AND [ 44] process for ingot weight between X2a - X2b tonnes
        AND      DO: Assign Variable
                 E_Process1$:="Double Slag Furnace"
        AND      DO: Assign Variable
                 E_Process2$:="Single Slag VAD degas"

      + OR   [ 38] not VCD treated steel
      +  AND [ 42] process for ingot weight between X3a - X3b tonnes
        AND      DO: Assign Variable
                 E_Process1$:="Double Slag Furnace"
        AND      DO: Assign Variable
                 E_Process2$:="2 Single Slag VAD/LF degas"

[ 50] Steelmaking procedure approval expert
      + IF   [ 34] Main Menu
      +  AND [ 57] the expert is choosing
        AND      DO: Restart Rule

        OR   [ 59] User chooses

[ 51] T-percentages being processed            Sp
        IF       DO: Assign Variable
                 per$:=left$(bar$,score/4)
      F AND      DO: View Form
```

CRYSTAL EXPERT SYSTEM
STEELMAKING PROCEDURE APPROVAL EXPERT

```
                    Éííííííííííííííííííííííí»
                     º Please wait ...      º
                     º                      º
                    Èííííííííííííííííííííííí¼


                     System processing rate:
                    º[per$                 ]º
                    Èííííííííííííííííííííííí¼
                    0         50        100
      COL : SURR      White on Blue
      COL : 0,0       Blue on Cyan      COL : 4,0       White on Black
      COL : 5,0       White on Blue     COL : 19,0      Lt_Gray on Black
      COL : 19,23     White on Blue     COL : 19,54     Lt_Gray on Black
      COL : 20,0      Blue on Cyan
      OUT : 17,25,49
         per$

[ 52] teeming temperature over Liquid temperature
    + IF   [ 54] Temp:=X2-1
      AND       DO: Assign Variable
                E_Temp:=X2-1

    + OR   [ 53] Temp:=X2-2
      AND       DO: Assign Variable
                E_Temp:=X2-2

[ 53] Temp:=X2-1
     IF        DO: Test Expression
               Steel$="No"

[ 54] Temp:=X2-2
     IF        DO: Test Expression
               Steel$="Yes"

[ 55] test if print selected 1                Sp
     IF        DO: Test Expression
               menuvar1=2
    F  AND     DO: Print Form

                      CRYSTAL EXPERT SYSTEM
               STEELMAKING PROCEDURE APPROVAL EXPERT


            The Steelmaking Procedure is not accepatable as
              the Expert System has evaluated below,

            [arrR1$[1]                ]
            [arrR1$[2]                ]
            [arrR1$[3]                ]
            [arrR1$[4]                ]
            [arrR1$[5]                ]
            [arrR1$[6]                ]
            [arrR1$[7]                ]

                  Press any key to exit
```

```
            COL : SURR      White on Blue
            COL : 0,0       Blue on Cyan        COL : 4,0       White on Blue
            COL : 4,35      White on Black      COL : 5,0       White on Blue
            COL : 5,26      White on Black      COL : 5,28      White on Blue
            COL : 19,0      Lt_Gray on Black    COL : 20,0      Blue on Cyan
            OUT : 9,16,61
                arrR1$[1]
            OUT : 10,16,61
                arrR1$[2]
            OUT : 11,16,61
                arrR1$[3]
            OUT : 12,16,61
                arrR1$[4]
            OUT : 13,16,61
                arrR1$[5]
            OUT : 14,16,61
                arrR1$[6]
            OUT : 15,16,61
                arrR1$[7]

        AND     DO: Test Expression
            output("SMP2")
        AND     DO: Quit

[ 56] test if print selected 2                  Sp
      IF        DO: Test Expression
                menuvar2=2
    F   AND     DO: Print Form

                        CRYSTAL EXPERT SYSTEM
                STEELMAKING PROCEDURE APPROVAL EXPERT


                The Steelmaking Procedure has been approved by
                                the Expert system

                [arrR1$[1]                  ]
                [arrR1$[2]                  ]
                [arrR1$[3]                  ]
                [arrR1$[4]                  ]
                [arrR1$[5]                  ]
                [arrR1$[6]                  ]
                [arrR1$[7]                  ]



                            Press any key to exit
            COL : SURR      White on Blue
            COL : 0,0       Blue on Cyan        COL : 4,0       White on Blue
            COL : 4,34      White on Black      COL : 5,0       White on Blue
            COL : 5,30      White on Black      COL : 5,32      White on Blue
            COL : 19,0      Lt_Gray on Black    COL : 20,0      Blue on Cyan
            OUT : 9,18,63
                arrR1$[1]
            OUT : 10,18,63
                arrR1$[2]
```

```
            OUT : 11,18,63
                arrR1$[3]
            OUT : 12,18,63
                arrR1$[4]
            OUT : 13,18,63
                arrR1$[5]
            OUT : 14,18,63
                arrR1$[6]
            OUT : 15,18,63
                arrR1$[7]

        AND     DO: Test Expression
                output("SMP1")

[ 57] the expert is choosing                    Sp
      + IF   [ 30] initail variables
      +  AND [  2] ask all the questions
      +  AND [ 51] T-percentages being processed
      +  AND [  7] assign a bar
      +  AND [  1] apply the answers to the questions
      +  AND [ 58] then display the results

       OR       DO: Succeed

[ 58] then display the results                  Sp
       IF       DO: Assign Variable
                score:=80
      +  AND [ 51] T-percentages being processed
      +  AND [ 21] display results
      +  AND [ 20] determine the next step...

[ 59] User chooses                              Sp

[ 60] user selects from the "what-next" menu
      F IF      DO: Menu Question  next

                        CRYSTAL EXPERT SYSTEM

            Select answer by moving the cursor right and left keys

                          Press Enter when ready
            COL : SURR     White on Blue
            COL : 0,0      Blue on Cyan      COL : 4,0     White on Black
            COL : 5,0      White on Blue     COL : 19,0    Lt_Gray on Black
            COL : 20,0     Blue on Cyan

[ 61] VCD treated steel
       IF       DO: Test Expression
                VCD$="Yes"

[ 62] Welcome screen
       IF       DO: Test Expression
                graphics("cks3.hi")

[ 63] what processes are supposed               Sp
      F IF      DO: Display Form
```

CRYSTAL EXPERT SYSTEM
STEELMAKING PROCEDURE APPROVAL EXPERT

For the conditions given abobe,
what are the processes supposed to be,

1. Double Slag Furnace
2. Single Slag VAD degas
3. Single Slag Furnace
4. 2 Single Slag VAD/LF degas
5. VAD Not Specified

ENTER YOUR CHOICES <P1> AND <P2>

Enter your choice numbers into those fields above

```
                Press Enter when ready
        COL : SURR      White on Blue
        COL : 0,0       Blue on Cyan        COL : 4,0       White on Blue
        COL : 20,0      Blue on Cyan
        IN  : 17,40,43,0 Blue on Gray
              P1
        IN  : 17,49,52,0 Blue on Gray
              P2

   AND      DO: Assign Variable
          P$[1]:="Double Slag Furnace"
   AND      DO: Assign Variable
          P$[2]:="Single Slag VAD degas"
   AND      DO: Assign Variable
          P$[3]:="Single Slag Furnace"
   AND      DO: Assign Variable
          P$[4]:="2 Single Slag VAD/LF degas"
   AND      DO: Assign Variable
          P$[5]:="VAD Not specified"
 + AND [  9] assign processes

[ 64]           CRYSTAL  MASTER  RULE        Sp
     IF        DO: Init. Variables
   + AND [ 33] Introduction
   + AND [ 50] Steelmaking procedure approval expert
[var] arr1$[16]

[var] arrR1$[10]

[var] INGOT
   VAL: INGOT>0&INGOT<120

[var] J
   VAL: J<=0&J<=9

[var] ladles$
   VAL: ladles$>"0"&ladles$<"5"
```

```
[var] liquid
    VAL: liquid>0&liquid<300

[var] metal
    VAL: metal>0&metal<100

[var] P$[10]

[var] P1
    VAL: P1>=1&P1<=10

[var] P2
    VAL: P2>=1&P2<=10

[var] Temp
    VAL: Temp>0&Temp<=200

[var] Wt
    VAL: Wt>0&Wt<=300

END.
```

APPENDIX 6

HYDROGEN DEGASSING PROGRAMME IN C++

```
// Filename: C1Hydro.cpp
// This program is for calculating the hydrogen degassing hours by calling an external function from
// FEL CAPP database system. It deducts degassing effects from each stage of heat treatment,
// transferring, equalising, heating up or cooling down, and so on.

#include <iostream.h>
#include <fstream.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <stdio.h>
#include <conio.h>

struct h_treatment              //global structure to store heat treatment cycles.

  {
        char process[20];       // process array to store heat treatment cycles.
        float temp;             // temperature variable.
        float rate;             // heating or cooling rate.
        float time;             // time required for individual process.
  };

struct ingot_info               //global structure to store ingot information.

  {
    float radius_ingot;
    float concen_i;
    float concen_c;
  };

//Prototype functions defined below.
struct ingot_info get_data_c(struct ingot_info &ingot_var);
float calcu_d(float temp);
void print_result(float temp, float time_final);

int main()

{

//Calculate di_t required for hydrogen degassing process.
  float temp=700;       //initial temperature at 700 °C.
  float temp1;          //temperature variable for each heat treatment cycles.
  int ctr=0;            //total items of processes.
  float time_final;     //time required for hydrogen degassing cycle.
  float time_req;       //total time required for hydrogen degassing.
  float diff_t;         //hold values for individual diffusion times time-required.

  h_treatment treatment[20];    //local structure variable array to read disk file records.
  ingot_info ingot;             //local variable to hold ingot radius, initial concentration and
                                //centre concentration values.
  ingot = get_data_c(ingot);    //read ingot radius, concen_i and concen_c from function
                                //get_data_c.
```

```
ifstream in_fh;
in_fh.open("Hydro.dat",ios::in);

if(!in_fh)
  {
  cout <<"Error opening file2.\n";
  exit(0);
  }
else
  {
  ctr=-1;
  do
   {
   in_fh >> treatment[ctr].process;
   in_fh >> treatment[ctr].temp;
   in_fh >> treatment[ctr].rate;
   in_fh >> treatment[ctr].time;

   }while (strcmp(treatment[ctr].process,"COOL_TO") != 0) ;
  }

in_fh.close();

int ctr1=0;

do
{
        if (treatment[ctr1].temp==treatment[ctr1-1].temp)
        //then, calculate the temp & time for heating up or cooling down.
        {
        cout<<"\n";
        treatment[ctr1].temp=(treatment[ctr1+1].temp+treatment[ctr1-1].temp)/2;
        treatment[ctr1].time=(treatment[ctr1+1].temp-treatment[ctr1-1].temp);
        treatment[ctr1].time=fabs(treatment[ctr1].time);
        //determines the absolute value of the treatment[ctr1].time.
        treatment[ctr1].time/=fabs(treatment[ctr1].rate);
        }

        ctr1++;

}
while (ctr1<ctr);

time_req=log(ingot.concen_c/(2.3 * ingot.concen_i)) ;
time_req=pow(ingot.radius_ingot,2) * time_req;
time_req/=-5.75;
diff_t=time_req * calcu_d(temp);
ctr1=0;

for (ctr1=0; ctr1<ctr; ctr1++)
  {
   if (strcmp(treatment[ctr1].process, "DEGAS") ==0 )
   {
   temp1=treatment[ctr1].temp;
   }
   else
```

```
        {
        temp=treatment[ctr1].temp;
        diff_t=diff_t-calcu_d(temp)*treatment[ctr1].time;
         }
      }
   temp=temp1;
   time_final=diff_t/calcu_d(temp);

   print_result(temp1, time_final);   //calls printing function to print result to the screen.
                                      //hold for uers to press a key to continue.

 printf("press any key to continue:");
 while(!kbhit());

 return 0 ;

 }

//*********************************************************************
//This function is to read ingot radius and hydrogen concentration values at the initial and
//the centre of the ingot from a DOS file which is created by DEase query.

struct ingot_info get_data_c(struct ingot_info &ingot_var)

 {
  ifstream in_fc;
  in_fc.open("Ingot-In.dat",ios::in);
  if(!in_fc)
   {cout <<"Error opening file1.\n";
   exit(0);
   }
   else
   {in_fc>>ingot_var.radius_ingot>>ingot_var.concen_i>>ingot_var.concen_c;
   in_fc.close();
   }
 return(ingot_var);
 }

//*******************************************************************
//This function is for calculating the diffusion value at each cycle of the heat treatment.

float calcu_d(float temp)

   {
   // C++ codes to calculate individual D_i required for each stage of treatments.
   float local_d;            //hold the diffusion values for individual temperatures.

   local_d=(0.93/(10*10*10))*exp(-2.7*1000/(1.987*(temp+237)));

   return(local_d);
   }

//*****************************************************
//This function is for printing the final result on the screen for users.

void print_result(float temp, float time_final)
```

```
{
int count=0;
for(count=0; count<15; count++)
{
 cout<<"\n\n";
}

if (time_final<=0)
{
printf("        ******************************************\n");
printf("        *                                        *\n");
printf("        *   For the given Anealing process,    *\n");
printf("        *   The component is safe degassed.    *\n");
printf("        *                                        *\n");
printf("        ******************************************\n");
}
else
{
printf("        ******************************************\n");
printf("        *                                        *\n");
printf("        * Time required for Hydrogen Degassing  *\n");
printf("        * can be           %.1f", time_final);
printf(" Hrs.        *\n");
printf("        * at the temperature: %.0f ", temp);
printf(" Degree C.   *\n");
printf("        *                                        *\n");
printf("        ******************************************\n");
}
for(count=0; count<10; count++)
{
cout<<"\n";
}
ofstream out_r;                         //declare file pointer to write final result.
out_r.open("time_r.dat",ios::out); //store results into DOS file ready to be called by
if(!out_r)                              //DEase quey.
{cout<<"Error opening time_r.\n";}
else
{out_r<<time_final<<"\n";
}
out_r.close();
return;
}
```

FINISH.