Computers ind. Engng Vol. 22, No. 2, pp. 133–146, 1992
Printed in Great Britain.

0360-8352/92 $5.00 + 0.00

THE DEVELOPMENT OF A PROTOTYPE DSS FOR THE DIAGNOSIS OF CASTING PRODUCTION DEFECTS

ROBERT T. PLANT and QING HU
Department of Computer Information Systems, University of Miami, Coral Gables, FL 33124, U.S.A.

(Received for publication 14 October 1991)

Abstract—The aim of this paper is to design a decision support system that allows the causes of casting defects to be determined. A prototype system is developed through a rigorous development methodology and illustrates a style of development that attempts to ensure system maintainability, correctness, consistency of deduction and promotes quality software. The system allows casting parameters to be set and illustrates diagnostic reasoning of casting faults made.

INTRODUCTION

Machine building industries—such as automobile manufacture, minine, metallurgy and aviation—are heavily dependent upon the creation and use of metal parts with complex internal and external shapes. The only possible approach to create many of these parts is to utilize the casting technologies. The process through which castings are created is in itself a complex and involved processes. Some of the major steps are illustrated in Fig. 1.

Each of these individual steps and their associated production parameters can be a source from which casting defects may originate: that is defects such as pinholes, metal penetrations, and cracks [1]. Any kind of casting defect will seriously influence the performance of the equipment or the component using the defective casting. For example, should one or more blades of an aeroplane's turbo fan engine, which are typical castings, have micro-cracks, then as the blade undergoes stress during operation, the micro-cracks will develop into macro-cracks, with potentially critical consequences. Therefore, in practical production, the casting production process is carefully controlled and casting defects are tightly monitored by inspecting each casting from the production line with the help of specially designed devices. If the defect rate is detected as above a given threshold, then this indicates that some production parameters are out of control. A team of experts on defect diagnosis is then called to examine which parameters in which step(s) are causing the defects. A number of adjustments can then be enforced in the production process to ensure that the defect will not occur again. However, the process is complicated because many of these factors are highly interrelated: that is a correction in one parameter value may inadvertently affect other parameters. The practical consequences are, rather than try to prevent all defects, the expert team tries to control the scrap rate of castings below a certain percentage, typically 5% [2–3].

All of these factors make defect diagnosis a daily task for the casting production engineers. The aim of this paper is to describe a prototype decision support system created to assist the engineers improve their production performance rates.

The advantages of Decision Support System (DSS) are such that they provide a source of expertise when not otherwise available, standardize deductions and act as a uniform repository of knowledge that can be updated in line with emerging or new techniques. These and other advantages have been extensively documented in the DSS literature [4–6].

SYSTEM SPECIFICATION

The domain can be seen, from our previous discussion, to be suitable for a knowledge-based solution. However, it can also be seen as extensive in scope; therefore, it is necessary for us to define

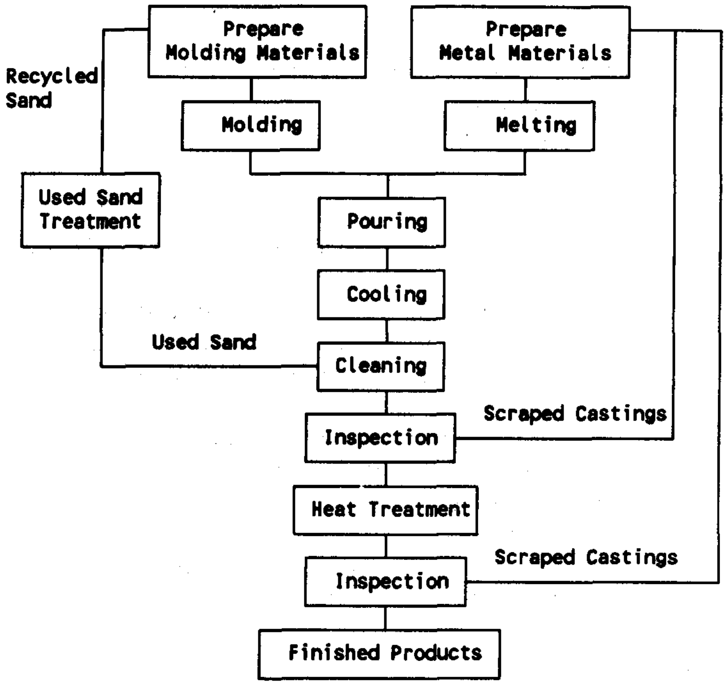

Fig. 1. Casting process.

the boundaries of our prototype system. This allows us to perform validation and verification procedures on the system [7].

We therefore outline the following three points of our specification constraints—area outline, problem specification, and system design.

Area Outline

In our DSS we simplify the domain to cover the following areas (see Fig. 2).

It is not designed to cover the areas of cooling, cleaning, used sand treatment or inspection processes.

The system can, however, be built up to cover these areas at a later stage, and the finer technical details can also be added to tune the system. The aim of the prototype was to create a system with a knowledge base of fundamental knowledge containing enough depth to allow the system to be tested in practical production situations.

Problem Specification

There are four parts to the problem specification:

Part 1: system parameters

The system has been designed to perform the following two major jobs in casting production for the domain defined above.

- For a given new casting product, select the best values of production parameters which will ensure the best possibility of sound castings.
- For a given defect, find the parameters which are major causes for this particular defect.

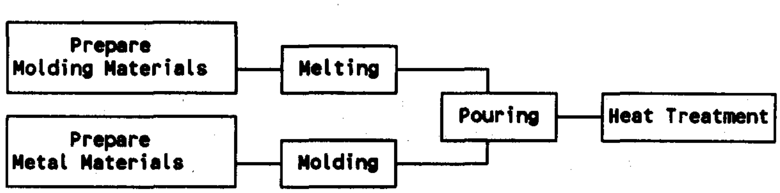

Fig. 2. Simplified casting process.

Part 2: casting parameters

The castings under consideration will be the following two classes:
- Grey iron castings (small, medium, large)
- Carbon steel castings (small, medium, large)

Part 3: defect parameters

A number of casting defects occur in daily production. The system will consider the following three kinds of defects:

Pinhole. It is a surface defect which occurs when gas evaporated from the molding materials invades into the casting surface during the liquid metal solidification process. This defect may also occur when the gas dissolved in the liquid metal is unable to escape into the air in the solidification process [3].

Metal penetration. This is also a surface defect which occurs when the sand molds are so porous that the liquid metal can penetrate into the molds or cores during the pouring and solidification process due to high liquid metal static pressure.

Cracks. Cracks may occur on the surface of castings or inside the body of castings. Cracks are the result of thermal stress created by the non uniformly changing temperature of casting metal during the solidification and heat treatment process [1].

Part 4: production parameters

Hundreds of parameters are involved in casting production and they all need to be monitored and controlled. The system in its prototype stage will only handle a subset of the full array of parameters:

Metal materials
- Condition of raw metal materials

Molding material
- Mold and core sands
- Average sand grain size
- Mold and core castings
- Mold and core binders

Molding mixture properties
- Binder weight percentage
- Mold hardness
- Compactability
- Permeability
- Gas evaporation

Melting and pouring
- Melting temperature
- Holding time
- Pouring temperature

Heat treatment
- Heating rate
- Keeping time
- Cooling rate

System Design

The design and development was such that it had to satisfy the following three requirements. Firstly, fulfil all the desired objects discussed previously. Having separate knowledge bases for each class of castings and defects, the system thus promotes data/knowledge integrity and maintenance. Secondly, the system has to be easy to use with a user friendly interface. Thirdly, casting production knowledge is rich in data and this necessitates that the system be able to access data in a database. The system will be able to store optimum production parameters within the database. The user responses to questions, which, in addition to system generated data, can be easily compared to

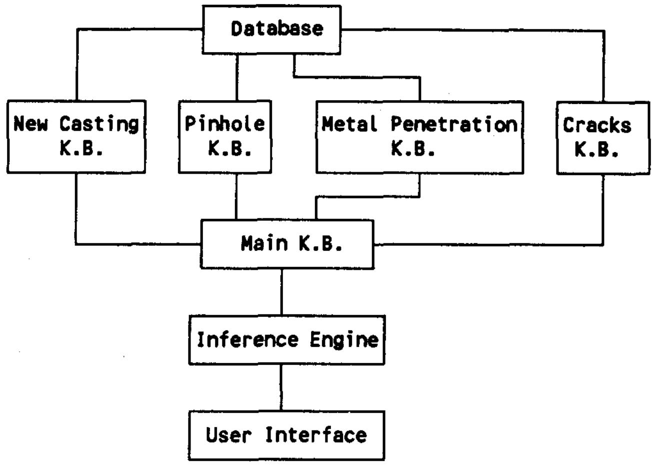

Fig. 3. System architecture.

those parameters. The system can then offer a solution. The architecture of the system is shown in Fig. 3.

SYSTEM DEVELOPMENT

The development of the DSS followed a methodology that attempted to promote rigor and accountability into the creation process [8]. The methodology can be simplified as shown in Fig. 4.

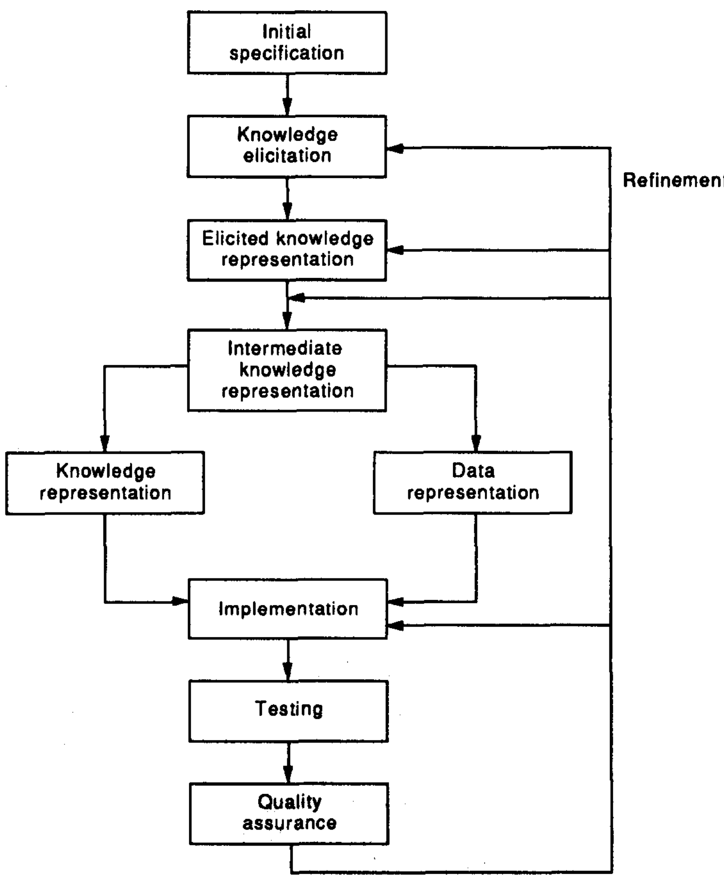

Fig. 4. Development methodology.

The knowledge engineer commences with a specification of the systems requirements. This is termed the *initial specification* because it is extremely difficult to fully specify knowledge-based systems in a formal manner. Thus, the developer attempts to create as rigorous specification as possible, in the style described and presented in the previous section. This specification is then used as a basis from which to proceed in system development. Its main functions are to define the boundaries of the systems domain, both in terms of breadth and depth, while acting as a baseline document so that the system developed can be compared against the initial specification requirements.

Having specified the system, the knowledge engineer then proceeds to select an *elicitation* technique [9] and extract the domain specific knowledge from the domain expert or knowledge source. The elicited knowledge is usually in the form of text, such as a transcribed interview, and this is known as the elicited knowledge representation. The third stage is to analyze the elicited knowledge, a process known as *knowledge* clarifying. The process may utilize intermediate representations with which to add structure to the knowledge e.g. decision tables or trees. The intermediate form allows the knowledge to select a representation e.g., rules [11], with which to implement the system. Finally, system *testing* and *quality assurance* measures can be performed. The step-wise development with multiple implementation independent stages, allows for errors to be easily corrected and gaps in the knowledge to be filled with consistency.

We will now consider each of these stages in the development of the casting decision support system.

Knowledge Elicitation

Knowledge elicitation is a process in which the domain knowledge is extracted from a domain expert or other sources and organized into a form that can subsequently be analyzed and used in the knowledge representation process. Several techniques are available to the knowledge engineer including reporting, interviewing and literature referral.

The knowledge elicitation processes used in this study included interviewing and literature referral. Interviewing can be (a) unstructured, (b) structured or (c) focused. In an unstructured interview, the knowledge engineer (after giving a few seed questions) allows the domain expert to develop the discussion, direct it to an area he feels is important. The knowledge engineer acts mainly to ensure that the domain expert does not digress too far from the area of interest.

Within the casting production problem domain, an unstructured interview would have the following form:

KE: Please tell me what you would do if it is reported that the scrapped casting rate increased over 2% in today's production?
DE: Well, first I would check what kind of defect is the major one to generate this out-rate increase. It may be pinholes, metal penetrations, cracks, nonmetallic inclusions...

In a structured interview, the knowledge engineer takes a much more leading role. The knowledge engineer attempts to regulate the depth of the knowledge the interview is generating in a more controlled way, introducing new information into the discussion when deemed necessary. Structured interviews undertaken for the casting system were of the following form:

KE: Please tell me what you would do if it is reported that the scrapped casting rate increased over 2% in todays production?
DE: Well, first I would check what kind of defect is the major one to generate this out-rate increase. It may be pinhole, metal penetration, cracks, nonmetallic inclusions...
KE: Let's assume that no human error occurred. If it's a pinhole, what would you do next?
DE: If it's a pinhole, I need to know what the kind of casting upon which the defect occurred most often, and then...

The knowledge engineer then would proceed taking the problem a stage at a time and asking the knowledge engineer to break down his deduction methods before examining each of these. The individual steps that the domain expert uses would most likely come under scrutiny of the knowledge engineer through a focused interview, where the level, scope and grain size of the information elicited is reduced from the general to the specific. For example, in the above interview, the knowledge engineer would ask the following questions:

KE: Why do you want to know the type of the casting
DE: Because steel casting and iron casting use different materials in the production process, therefore different parameters have to be taken into account.
KE: What kinds of special materials are used in the steel casting production? Are they special to create pinhole defects?
DE: Steel castings have a higher melting point than iron castings; they, therefore, require higher grade refractors . . .

The use of interviewing was supplemented by references to the literature. This was an important technique at all stages because technical details, terms and interactions not detailed by the domain expert were needed.

Developing the Representations

As noted, the result of the knowledge elicitation phase is a series of textual natural language transcripts (English and Chinese in this study). The use of these voluminous texts from which to directly implement the system is precluded by their inherent ambiguity, noise inconsistencies, incompleteness and scale. The text, therefore, has to be refined into a less ambiguous form that promotes completeness and consistency. The form we advocated to use was that of the decision table.

Intermediate representation: decision tables

The casting process involves many factors that have to be included in the conditions stub of the decision table. This, however, makes a single decision table so large, that any attempted gains in completeness and correctness are precluded. For example, to find out what causes the pinhole defect, at least eight factors must be taken into account before a defect diagnosis can be made and an action taken. The total possible number of conditions being 256 (Table 1). In order to overcome the problem of representing and manipulating very large decision tables we decided to partition the large decision table into a hierarchy of smaller ones, where each table focused the decision process towards a specialized area or action. For example, at the highest level we can decide whether the materials are raw metals or not, having decided this we can then pass control to two further tables, one focusing upon raw metals the other none raw metals. These tables then draw

Table 1. Complete decision table

	$2^8 = 256$					
Raw Metal Materials	Y	Y	Y	Y	......	N
Melting Temperature	Y	Y	Y	Y	......	N
Holding Time	Y	Y	Y	Y	......	N
Pouring Temperature	Y	Y	Y	Y	......	N
Permeability	Y	Y	Y	Y	......	N
Binder Type	Y	Y	Y	N	......	N
Binder Percentage	Y	Y	N	Y	......	N
Water Percentage	Y	N	Y	Y	......	N
Actions						
Action 1		X				X
Action 2			X			X
: : Action n					......	X

Table 2. Raw metal decision table

Conditions					Group Factor
Clean	Dry	Dirty	Rusty	Wet	Raw_Metal
Y	Y	N	N	N	Y
-	-	Y	-	-	N
-	-	-	Y	-	N
-	-	-	-	Y	N

and group together only those factors involved in refining the diagnosing of defects within their area of focus. Through this approach, the total number of decisions for a particular problem is remains the same, yet the complexity at each stage is greatly reduced, making them easy to create, interpret and use. For example, Table 2 is used to indicate whether a further decision table that considers defects in the raw metal should be utilised. This would be appropriate for example when the raw metal was clean, dry and not rusty, dirty or wet. This is represented in line one of the decision table.

Table 3 is used in a similar way to partition the problem into the problems associated with liquid metal.

Having decided that the defect is in the area of raw metals then this can be focused upon as shown in Fig. 4 such that a diagnostic action can be taken.

Knowledge representation: rules

Knowledge representation schemes describe in terms of data structures the knowledge structures used by the expert over which his deductions occur. The question of how knowledge is represented within an expert or decision support system is of central concern. This is because the structure determines the type and ease of reasoning that can occur over a given knowledge base, ultimately determining the capability of the system.

A number of techniques are used to represent different knowledge types and the interrelationships of that knowledge: (i.e. frames, semantic networks, production systems, logic [11]). We decided to utilize a production system architecture [12] for our system, due in part to the following reasons: the structure of the casting defect diagnostic knowledge is suitable to being represented in a rule form, production systems are easy to implement, understand and use; plus, the modularity of production systems provide flexibility in the development and maintenance of the knowledge base. The use of a production system representation also allows for the decision tables to be easily transformed into rules, thus maintaining semantic consistency.

An example of a rule from the decision table in Table 4 is of the form:

```
RULE 1
  IF      raw_metal= yes AND
          binder_type= no AND
          permeability= yes AND
```

Table 3. Liquid metal decision table

Conditions			Group Factor
Met_Temp	Hol_Time	Pou_Temp	Liquid_Metal
Y	Y	Y	Y
N	-	-	N
-	N	-	N
-	-	N	N

Table 4. Reduced decision table

Raw_Metal	Y	Y	Y	Y	Y	Y	Y	Y
Binder_Type	Y	Y	Y	N	N	N	Y	N
Permeability	Y	Y	N	Y	N	Y	N	N
Gas_Evaporation	Y	N	Y	Y	Y	N	N	N
Actions								
Action 0	X							
Action 1						X		
Action 2				X		X		X
Action 3			X				X	X
Action 4		X				X	X	X

```
           gas_evaporation= yes
    THEN   action= 2
Other rules are of the form
  RULE 21
    IF     cbinder= sodium_silicateAND
           sodsi >= (sodsil) AND
           sodsi =< (sodiu)
    THEN   ccompac= yes
    ELSE   ccompac= no
    BECAUSE"Too much or too little sodium silicate
              in sodium silicate sand will result in loose
              cores made from it, which in turn makes the
              liquid metal penetrate easily".
```

This shows how explanations can be attached to rules, allowing the system to inform the user of the systems reasoning strategies. This is an advantage that production systems exhibit. The rule structure also allows the use of "what if" experimentation on the part of the user and allows the user to change the parameters of a problem and examine the consequences.

Data representation: relational database

As we have previously mentioned, the casting process is a data intensive domain and we need a representation in which the data can be conveniently represented, refined and updated. The mechanism chosen for this was that of a relational database.

NAME	Mold_Sand	Core Sand	M_SS_L_LT	M_SS_U_LT	C_SS_L_LT	C_SS_U_LT
Small_Steel_Casting	Silica II	Silica I	75	85	95	105
Medium_Steel_Casting	Silica I	Silica I	55	65	80	100
Large_Steel_Casting	Chromite	Zircon	45	55	75	95
Small_Iron_Casting	Silica II	Silica II	80	95	100	120
Medium_Iron_Casting	Silica II	Silica II	70	90	90	110
Large_Iron_Casting	Silica II	Chromide	55	75	90	110

Fig. 5. Molding material data base.

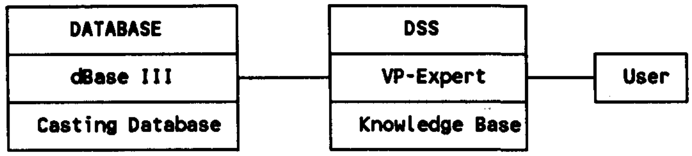

Fig. 6. System design.

An example of a set of relations that describe the parameters associated with casting mold materials is illustrated in Fig. 5.

Figure 5 shows, for example, that the molding sand type and core sant type necessary for casting a small steel casting should be silica II and silica I respectively. Further, it details that the molding sand size should be between a lower limit of 75 (M_SS_L_LT) and an upper limit of 85 (M_SS_U_LT) and that the casting sand size should be between a lower limit of 95 (C_SS_L_LT) and an upper limit of 105 (C_SS_U_LT). These sizes are for the average grain size, based upon the American Foundry Societies basis of calculation.

Implementation

The knowledge engineer having acquired the domain knowledge and data and having represented that information in forms that would facilitate retrieval of knowledge-based decisions, could then implement the system. This was accomplished through two components: a knowledge-based component and the database component. The system architecture presented in Fig. 3 was implemented through use of an expert system shell, VP-Expert [13, 14], which interfaced with a dBase III database [15] (see Fig. 6).

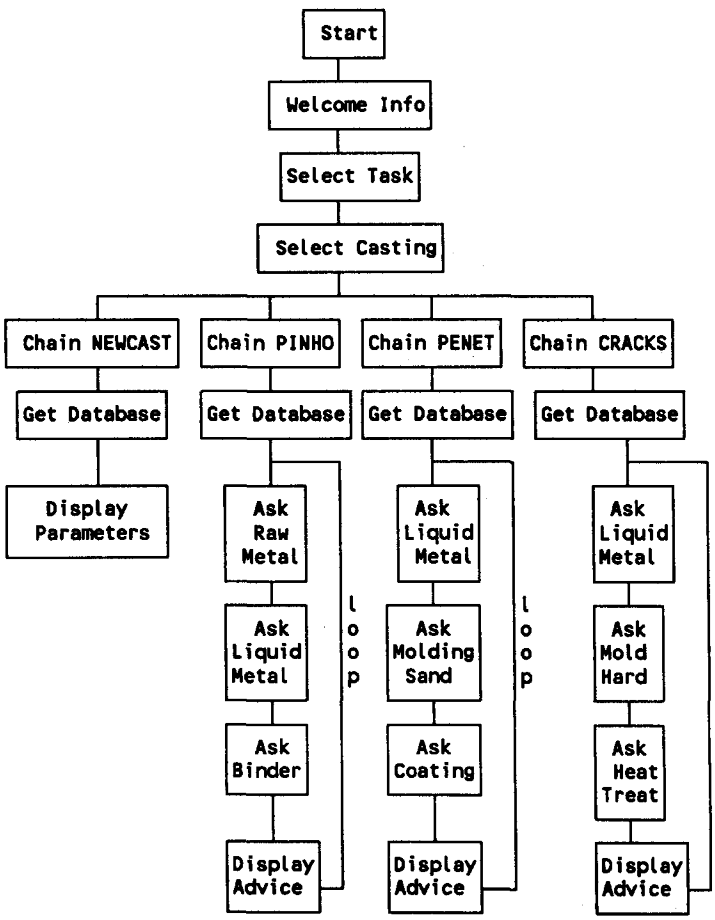

Fig. 7. System logic flow chart.

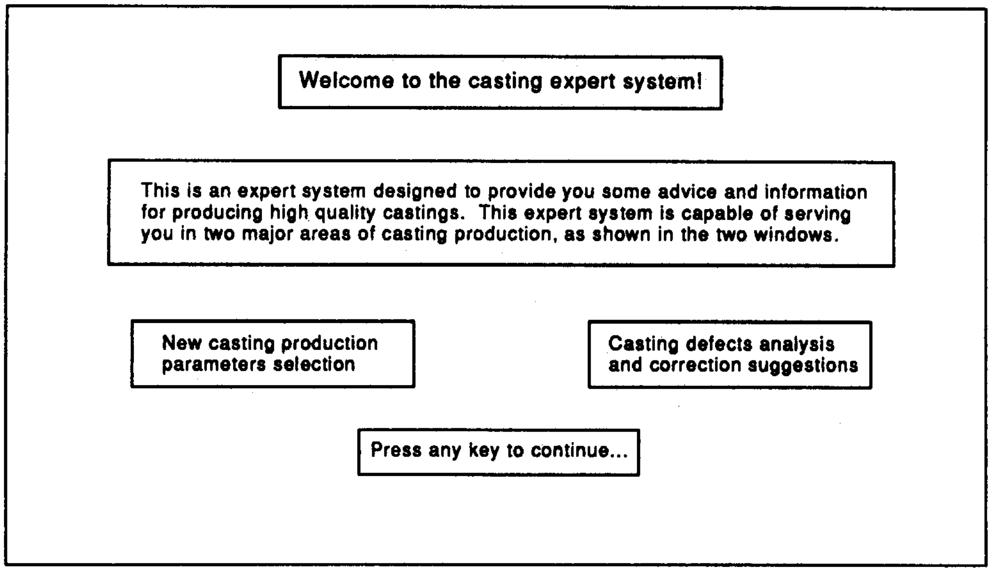

Fig. 8. Introductory screen.

The implementation of the system was performed with system maintenance and upgrading in mind and so extensive use of partition of both the knowledge base and database were made, thus increasing the modularity of the system.

A simplified logic, is illustrated in Fig. 7.

The system logic flow chart given as Fig. 7 shows how different problem types chain the system to different parts of the modularized data or knowledge base. This was found to be an effective implementation strategy which facilitated modification.

SYSTEM OPERATION

The casting decision support system was designed to be user friendly and to require as little interaction as possible, thus enabling a wide user group to take advantage of the system and to minimize the potential for input error. After the initial introductory screens of instruction, the user is asked to input data and information as the system deems necessary.

The initial selection data the system needs is whether the user wishes assistance in the selection of casting parameters or in casting defect analysis (Fig. 8).

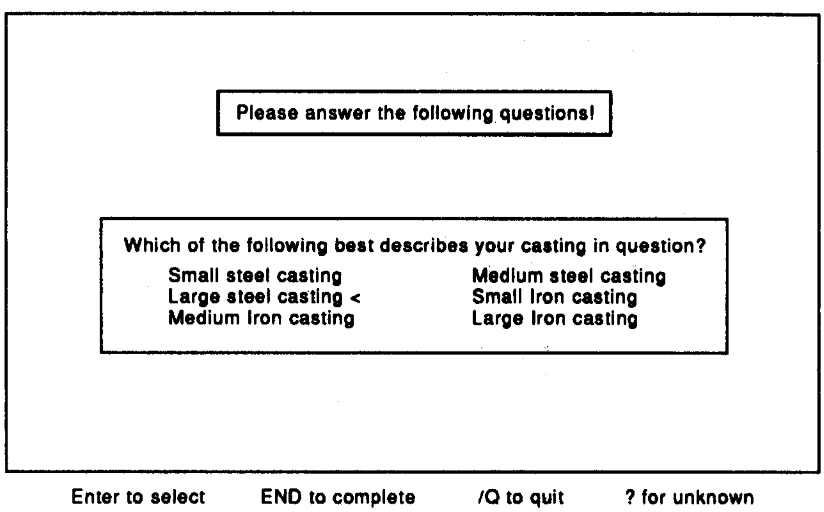

Fig. 9. Casting category screen.

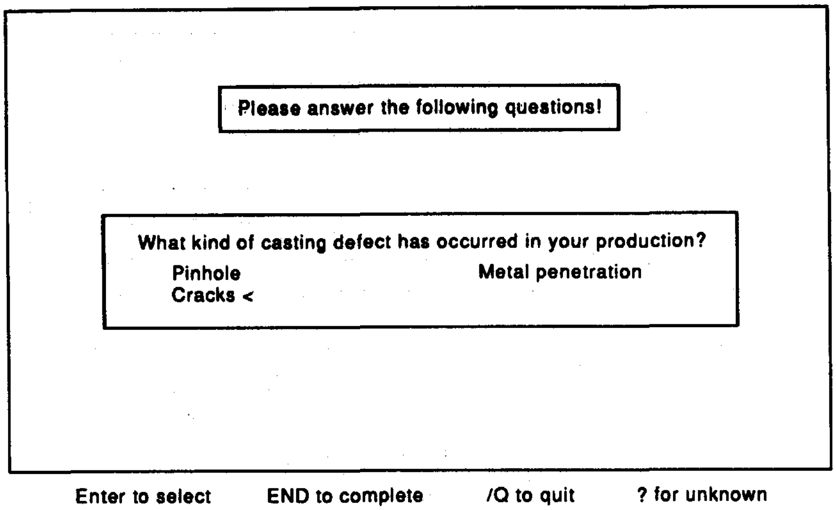

Fig. 10. Casting defect screen.

We now show a user interaction after the system has been asked for assistance in casting defect analysis, First, the casting category has to be determined (see Fig. 9).

The type of casting defect is then determined (see Fig. 10).

Having determined the basic data the system then focuses upon providing assistance in that area (see Fig. 11).

This is achieved through a series of detailed questions relating to the parameters of the production process (e.g. see Figs 12, 13 and 14).

At the end of this process, the system delivers a diagnostic analysis of the form shown in Fig. 15.

TESTING AND QUALITY ASSURANCE

The promotion of quality in our syustem was a prime concern from its conception, and even though it was intended to be a prototype system, this did not give license to allow for poor design and implementation. Thus was the reasoning behind the use of the rigorous development

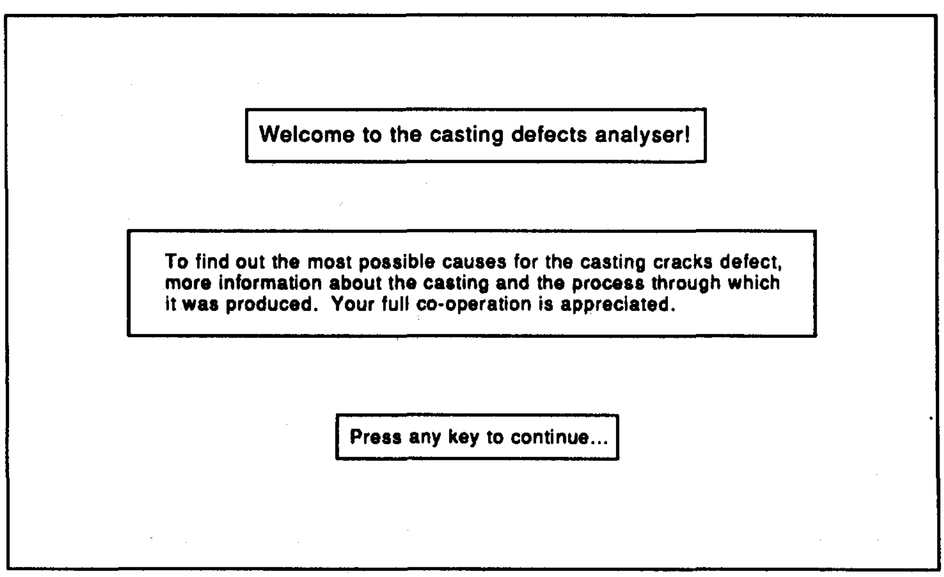

Fig. 11. Casting defect analyzer introductory screen.

Please answer the questions shown in the window!

At what temperature was the casting alloy melted?
1700

What is the value of mold hardness index?
6

Enter to select ? & enter for unknown /Q to quit

Fig. 12. Defect data entry screen.

Please answer the questions shown in the window!

What kind of mold sand was used in the defect area?

Silica I <	Silica II
Silica III	Chromite
Zircon	

Enter to select END to complete /Q to quit ? for unknown

Fig. 13. Defect data entry screen.

Please answer the questions shown in the window!

What is the value of temperature increasing rate when the heat treatment began?
1200

Enter to select ? & enter for unknown /Q to quit

Fig. 14. Defect data entry screen.

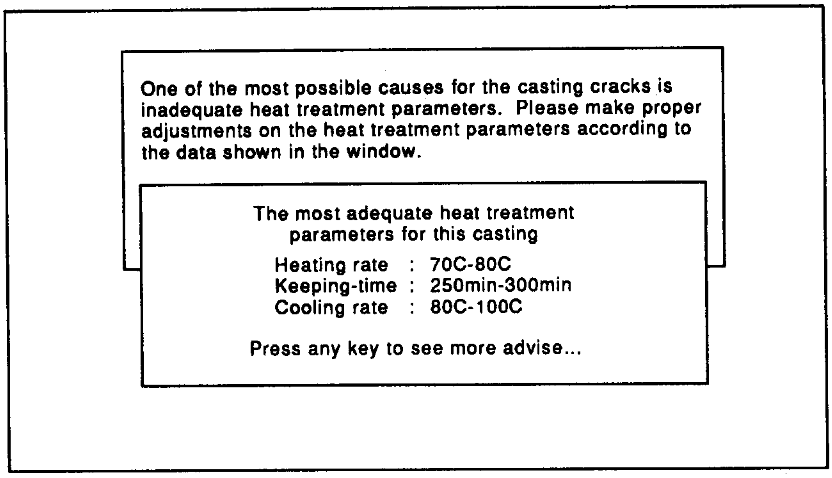

Fig. 15. Diagnosis results screen.

methodology outlined earlier. The use of this approach increased the three major factors effecting knowledge-based systems quality: consistency, completeness and correctness. The modular approach to development in conjunction with a stringent initial specification requirements has made the prototype extremely robust within its domain parameters.

The process of validation and verification in relation to knowledge-based systems has been demonstrated to be a significant problem [7, 16]. However, the techniques used in the development of our system are such that a high level of correctness is reached. This can be justified by exhaustively showing that the systems performance matches the requirements of the decision tables, a testing mechanism that is not normally feasible to demonstrate. The subsequent successor to this system will require alternative testing techniques such as critical data testing, random data tests or functional testing [17].

In order to ensure the systems validity, it was also tested against a human expert from the Shenyang Research Institute of Foundry in the People's Republic of China, who again attested to its validity within the boundaries of the domain specification.

SUMMARY AND CONCLUSIONS

The aim of this study was to produce a prototype decision support system that was capable of acting as an assistant to those studying casting defects. The system demonstrated that artificial intelligence techniques can be used in data and knowledge intensive environments when the system is supplemented by a database. The key to the management of this information is the availability of an environment that possesses a suitable interface between the knowledge base, the database and the inferencing mechanism. The study found the system architecture adequate for a prototype; however, a faster response time will be required in the next system of a larger scale.

The development benefitted significantly from the use of a rigorous methodology and the utilization of partitioned decision tables, both of which added to the correctness of the system.

The future development of this system will incorporate an expanded knowledge base that focuses upon exception data: for instance, where there may be deviations from the normal casting responses due to non standard situations. The system will also include the use of certainty factor algebras; this, however, may force the use of an alternative implementational environment due to the theoretical inadequacies of the certainty factor used in VP-Expert.

The system can easily be reimplemented due to the implementational independent nature of the knowledge and data representations created in the knowledge acquisition phase, another benefit of the adopted methodology.

The system in an expanded form could be used as an aid in the industrial workplace. It currently acts as a vehicle to educate casting technicians and helps develop an awareness of potential problems and solution strategies.

Acknowledgements—The authors wish to thank and acknowledge the constructive comments from the reviewers of this paper.

REFERENCES

1. C. F. Walton and T. J. Opar. *Iron Castings Handbook*. Iron Castings Society (1981).
2. J. M. Greenhill. Some important aspects of technical control in the production of high-quality grain-iron castings. *Foundry Trade J.* **Oct.** 737–771 (1979).
3. R. W. Heine and P. C. Rosenthal. *Iron Castings Handbook*. Iron Castings Society (1955).
4. P. G. W. Keen and M. S. S. Morton. *Decision Support Systems: An Organizational Perspective*. Addison Wesley, MA (1976).
5. S. L. Alter. *Decision Support Systems: Current Practice and Continuing Challenges*. Addison Wesley, MA (1980).
6. E. Turban. *Decision Support and Expert Systems*. Macmillan, NY (1990).
7. R. T. Plant. On the verification, validation and testing of knowledge-based systems. *Heuristics: J. Knowl. Engng*, **3,** 59–67 (1990).
8. R. T. Plant. A software engineering approach to the development of knowledge bases. In *Software Engineering and Artificial Intelligence* (Edited by D. Partridge). Ablex Press (1991).
9. A. M. Burton, N. R. Shadbolt, A. P. Hedgecock and G. Rugg. A formal evaluation of knowledge elicitation techniques for expert systems: domain 1. In: *Research and Development in Expert Systems IV. British Computer Society Series.* (Edited by D. S. Moralee). Cambridge University Press, Cambridge (1987).
10. M. Welbank. A Review of knowledge acquisition techniques for expert systems. British Telecom Labs Report, Marthlesham Consultancy Services, New York (1983).
11. D. A. Waterman. *A Guide to Expert Systems*. Addison Wesley, New York (1986).
12. C. W. Holsapple and A. B. Whinston. *Business Expert Systems*. Irwin, IL (1987).
13. R. Hicks and R. Lee. *VP-Expert for Business Applications*. Holden Day, CA (1988).
14. D. V. Pigford and G. Baur. *Expert Systems for Business: Concepts and Applications*. Boyd and Frazer (1990).
15. J. Carrabis. *dBase III Plus*. McGraw-Hill, New York (1987).
16. D. M. O'Leary. Methods of Validating Expert Systems. *Interfaces* **18,** 72–79 (1988).
17. J. Rushby. Quality measures and assurance for AI Software. NASA Contact Report NASI-17067 (1988).