An Investigation of the Feasibility of a CBR Approach to the Design of Sand Castings in the Foundry Industry

Tony Mileman

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

August 2002

Declaration

The work and results in this thesis are the sole work of the candidate and have not been presented as part of another award.

Abstract

The research described in this thesis investigates the feasibility of automating a key aspect of the Methods Engineering process of designing sand castings in the foundry industry by a case-based reasoning (CBR) approach.

The research answers four basic questions. These have been organised into a primary question and three subsidiary questions. The primary question asks: *Is it possible to automate the retrieval processes that a Methods Engineer uses in recalling analogous design cases for a given target design case by using a component decomposition of a shape derived from knowledge acquisition?* From this, the three subsidiary questions investigate the mechanism for component decomposition, the formulations of appropriate similarity metrics of components and finally, evaluating metric performance.

Methods Engineering is introduced and the investigation is focussed on an existing process of shape componentisation of 2-D section slices into elementary components corresponding to the way Methods Engineers reason about solid objects. A novel contribution of this research is the identification, abstraction and formulation of this procedure, and its automation by computer software. A further contribution is made by providing a series of similarity metrics for shapes in the casting domain; these include both metrics based on feature-value pairs, and metrics based on graph matching. A set of performance measures appropriate to the casting problem was formulated and a prototype retrieval system was produced and implemented as a CBR system. A test case base representing a sub-domain of rotationally symmetric castings was constructed and performance figures show that the system comes close to expert retrieval performance. This thesis concludes with a section regarding new research questions that emerge from the research and an agenda for future work that these questions give rise to.

Acknowledgements

I would like to thank my supervisors, Professor Brian Knight and Dr Miltos Petridis for their continued advice and support during the work and writing of this thesis. They have had to put up with a great deal from me and I'm grateful for their patience.

I would also like to thank Keith Preddy, for assistance in the research and acting as a consultant.

Dr Don Cowell also deserves special thanks for his help and for numerous discussions concerning the project in its early stages. Thanks also to Dr Patrick Mejasson for providing valuable software assistance.

Finally, and most importantly, I wish to thank my mother, Anne Mills, for her continued encouragement, support and a million other things.

Dedication

To My Family and Friends

Contents

Chapter 1 Introduction

Castings can be difficult to get right. Creating things never is easy.

John Campbell in introduction to *Castings* (Campbell 1991)

"Ever since humans have made artefacts, they have used design knowledge as design sources for new designs."

Myung Yeol Cha and John S. Gero (Cha and Gero 1998)

This chapter describes the research questions addressed, project objectives, how the questions have been answered and the contribution to knowledge made by the research. The research is concerned with automating a key aspect of the process of designing metal castings by retrieving past casting designs using a case-based reasoning (CBR) approach.

Casting is in general the cheapest process for the mass production of shaped metal components. However, there are problems of ensuring quality, and this depends on the existence of casting design knowledge. An important function within the foundry industry is the Methods Engineer, who uses a process called *methoding* to ensure that a finished casting supplied to the customer is of the highest possible quality; this is dependent on the skill and experience of individual Methods Engineers. The term *Methods Engineering* is defined as a human engineering process in a foundry context that ensures that a casting supplied to a customer is of the quality required and meets customer specifications within previously estimated costs. The process is essentially concerned with the design of the casting process; that is, filling the mould with molten metal, and the ensuing freezing (*solidification*) of the metal.

Methods Engineers consider a variety of problems associated with the process of solidification, the most crucial consideration being the shrinkage of metal that occurs as the metal freezes. The process of methoding is an iterative one, in which tentative designs (usually on large paper drawings of casting sections in 2-D, or increasingly as 3D computer-aided design CAD models) are evaluated, sometimes using simulation software of metal freezing in the mould (O'Halloran 1995).

The Methods Engineering process starts initially with a dialogue with the client, where possible re-design of the shape for a more efficient casting may be suggested. Modifications are usually made to an initial engineering drawing until the Methods Engineer is certain to the best of his/her knowledge that the design is *sound* (castable) and free from design problems.

It became apparent during this research that Methods Engineers remember and reuse previous casting designs when manually methoding new designs; in effect they solve new methoding problems by remembering previous ones. The aim of the research is to examine a way in which an essential aspect of Methods Engineering, that is the process of solving new methoding problems by retrieving past designs, can be modelled and automated by computer software. This was accomplished by identifying a special component decomposition technique that corresponded to the way Methods Engineers reason about solid objects. The term *componentisation* is defined as a process of systematically breaking a shape down into an assembly of elements, so that a simplified model of a shape's metal solidification can be assessed; its so-called *modulus model*. Extensive knowledge of casting is also associated with this shape decomponentisation; for instance, due to heat radiation junctions should be tapered.

The Methods Engineering process involves the aforementioned shape decomposition and the reuse of casting design knowledge that is linked to the shape decomposition in methoding a new design.

In this research, one approach to automate this pivotal aspect of the Methods Engineering process by software was implemented, and a prototype system was evaluated that can advise a Methods Engineer on the design aspects of a new casting by reference to similar castings that have been designed and tested in the past. The research answers a number of academic questions, which are detailed next.

1.1 Research Questions

Answers to the following questions constitute the contribution to knowledge made by the research.

1.1.1 Primary Question

The primary question posed and answered by this research is:

Is it possible to automate the retrieval processes that a Methods Engineer uses in recalling previous analogous design cases for a given target design case?

The aim of the research is to answer this question. The main problem is how to retrieve design cases, where the retrieval must be based on shape. Although there are many possible search indices, for example, the type of casting alloy, weight and general qualitative description (such as wheel, sea-gland, valve, engine bearing cap), these descriptions are too general for accurate retrieval.

It became apparent during the knowledge elicitation carried out in this research that a decomposition of shapes specific to the casting industry already existed in practice and a unique representation could be derived from this traditional decomposition. The research described in this thesis uses a graphical representation of shapes based on this decomposition as a foundation for shape retrieval within the simpler problem domain of rotationally symmetric castings.

Using this approach, a shape is made up of a number of horizontal and vertical 2-D cross sections, with each section slice decomposed into a set of joined components. The decomposition is a fundamental one used over many years by Methods Engineers, and it is based on a set of component types of significance in casting design.

Why this research question is important is explained in more detail later in the thesis overview section of this chapter, but briefly, the foundry industry is looking for software tools to assist with casting to reduce costs and to improve quality.

1.1.2 Revised Question:

The primary question could be revised in the light of initial research showing that a component decomposition was possible. The revised question asks:

Is it possible to automate the retrieval processes that a Methods Engineer uses in recalling previous analogous design cases for a given target design case by using a component decomposition of a shape derived from knowledge acquisition?

To answer this, the question can be broken down into a number of subsidiary questions, each of which needs answering.

1.1.3 Subsidiary Questions

The first of these subsidiary questions is:

What is the mechanism for component decomposition?

There are two choices for componentisation. Either a new set of criteria can be developed, or there can be an investigation as to how human practice can be emulated using the fundamental component types. By taking the first approach, an arbitrary division is used (see Natarajan *et al* 1989), which subdivides a shape into a uniform grid of cells. This is a technique utilised by other researchers, but so far has not provided new tools of worthwhile importance. Furthermore, this approach is unrelated to the experience of Methods Engineers.

There is a strong rationale for the alternative, that is, to emulate the shape decomposition scheme that is specific to Methods Engineering, especially since so much experiential knowledge has grown up around such decompositions. In addition to this, knowledge of re-design is often keyed to this element classification.

This question spawns the sub-problem concerned with the abstract representation of casting shapes. Ideally, a unique and comprehensive abstract data representation that is capable of supporting human reasoning of casting is required. The guiding principle that was used to direct this abstraction was to emulate the way a human expert performed the Methods Engineering process.

During the knowledge acquisition stage of the research, it was noticed the expert referred to 2-D sections of vertical and horizontal cross-sections. Effectively, the problem can be reviewed as one in which the componentisation is represented as an assemblage of elementary 2-D section slices, corresponding to the way Methods Engineers reason about solid objects. In this research however, the focus is on evaluating retrieval performance for individual section slices; that is, confining each shape as having one section slice. The problem of *n* slice retrieval is assigned for future research and this is investigated in Chapter 7 - Conclusions.

The next subsidiary question is:

How can appropriate similarity metrics of shape sections be formulated?

There is not a way to represent shapes uniquely and thus a technique of determining shape similarity is required. To achieve this, a number of features can be extracted from a shape section. Some of the most important features that can be identified include the following:

i. The types of components that make up a shape (for example, Bar and T-junction)

ii. The component assembly (that is, how a shape is assembled as a network of interlocking components)

These and other distinguishing features can be used to form different similarity measures that represent different case retrieval mechanisms. The features can also be combined to form a general similarity metric.

The final subsidiary question is:

How can performance be measured?

Of great importance in devising metrics is to evaluate the efficacy of performance. The best optimum procedure for this is to assess the outcome against human expertise, preferably a skilled Methods Engineer who has many years of foundry experience in casting design.

1.2 Contribution to Knowledge

This section briefly looks at the main achievements of the research in investigating the questions posed at the outset. The investigation focussed on an existing process using componentisation of 2-D section slices into elementary components corresponding to the way Methods Engineers reason about solid objects. A novel contribution of the research is the identification, abstraction and formulation of this shape decomposition approach and its evaluation. This resulted in a number of contributions both to the specific field of CBR in casting, and to CBR in general. These are:

1. A series of similarity metrics for shapes in the casting domain have been proposed. These include both metrics based on feature-value pairs, and similarity based on graph matching.

2. A set of performance measures appropriate to the casting problem has been formulated.

3. A prototype retrieval system has been constructed.

4. A test case base representing a sub-domain of rotationally symmetric castings has been constructed. Performance figures have been calculated for:

a) a human expert
b) automatic retrieval
c) optimum retrieval

It is possible from these performance figures to identify performance due to the case base and performance due to the retrieval. It is shown that automatic retrieval comes close to expert retrieval performance.

1.3 Overview of Thesis

The next chapter establishes the domain of the research, which is the casting industry. It reviews relevant literature regarding the research, including a background to casting and the problems of Methods Engineering. The foundry industry represents an important sector of the UK manufacturing industry, and there are many benefits to be obtained in improving good Methods Engineering practice by reducing reject rates and improving casting quality. Various approaches for modelling the expertise of Methods Engineers is investigated, including traditional numerical modelling and Artificial Intelligence (AI) techniques of rule-based expert systems and CBR. From a purely pragmatic view, the industrial problem is the provision of a system that a Methods Engineer will find useful during the methoding task, with the goal of leading to improvements in casting procedures. Prior work at Greenwich [Cowell *et al* 1993, 1994; Knight *et al* 1995] focused on developing a software tool that could construct a so-called modulus model[1] of the shape to be cast. This initial prototype was too limited to be of practical use, but pointed the way to the possibility of a large-scale prototype, forming the impetus for this research.

[1] *Modulus Model*. The solidification time of a shape is proportional to the square of its volume to cooling surface area. This ratio is referred to as the geometric modulus, or just the modulus.

Chapter 3 investigates knowledge elicitation and Methods Engineering, pinpointing that shape is of prime importance in methoding. Knowledge to ensure soundness (that is, quality castings) is derived from previous experience with castings of similar casting designs. A critique of existing approaches used to model shapes is given before justifying that Methods Engineers use familiar component elements with well defined moduli (using the above mentioned modulus model) and that casting shapes are represented in the language of the Methods Engineer.

Chapter 4 investigates the abstract representation of shapes based on the Methods Engineering shape decomposition as discussed in the previous chapter; this scheme is examined further in which 2-D cross sections are used to represent casting shapes from which similarity metrics were abstracted. This corresponds to the way Methods Engineers reason about solid objects. The representation of casting shapes used in the research is that of a graph structure; the problems of devising similarity metrics, which operate on graphs delineating casting shapes, is detailed in Chapter 5.

The retrieval of casting shapes by developing search algorithms for close partial matches is the problem addressed in Chapter 5. Similarity measures are based on features extracted from the structural graphs. Perfect similarity between two shapes is obtained when they have identical structural graphs. However, for graphs that do not match completely, there are a number of features that can be extracted and compared, and these are used to form search indices. A number of similarity metrics for shape retrieval are devised and critiqued.

Chapter 6 then goes on to cover the evaluation of these similarity metrics. The efficacy of various retrieval procedures was evaluated against a benchmark provided by a skilled Methods Engineer. The purpose of a Methods Engineer's retrieval is to decide on a number of essential factors. For the purpose of this research, these are taken as:

- Orientation (moulding direction) of the shape
- Number and position of feeders and chills
- Design Advice

A prototype system capable of retrieving cases for given target shapes, and able to predict the correct orientation and position of chills[2] and feeders[3] was implemented and tested using a suitable subdomain. It was decided to restrict the prototype system to a class of castings with rotational symmetry: such as wheels, armatures, and cylinders. This domain is coherent from a practical point of view in that it can be covered with a limited case base. However, the domain is sufficiently varied to encompass a wide range of casting problems.

Chapter 7 concludes with a summary of the research results and contributions of this thesis, along with an agenda for future research.

Seven appendices are provided in addition to the main body of this thesis. Appendix A and B have two conference papers relevant to this research presented by the PhD candidate at two international casting conferences, with a journal paper following these in Appendix C. Appendix D contains Visual Basic code for the Maximum Common Subgraph metric as discussed in Chapter 5 - Section 5.1.2 of this thesis. Appendix E contains AutoLisp code for the Component metric as discussed in Chapter 5 - Section 5.1.1 of this thesis. Appendix F contains details about the software implementation of the prototype CBR system and associated algorithms, including ones for finding the optimum weights settings. Appendix G contains a glossary of casting terms used in this thesis.

[2] *Chills* are heat-absorbing blocks embedded in the mould and mouldable materials that can force parts of the shape to freeze more quickly than neighbouring casting sections.
[3] *Feeders* are reservoirs that can supply molten metal to elements of the shape to account for the volume loss as a result of shrinkage occurring as the casting alloy freezes from liquid to solid.

Chapter 2 Literature Review

The aim of this chapter is to establish the domain of the research, which is the casting industry. Casting has a rich history, and this chapter maps out some of the major milestones in its six thousand year chronicle pertinent to this research. A key focus of this research is related to the task of the Methods Engineer, who plays an important role in the foundry industry. The Methods Engineer ensures quality castings by a process called *methoding*. For many years foundry practice was considered as a 'black art', and it is only in the past few decades that mathematical and computational techniques, particularly using numerical modelling and AI have been applied as computational assistants for the production of better quality castings. A software taxonomy is introduced, presenting the context of the current research as an heuristic process. The importance of this new heuristic approach for shape retrieval is investigated in the following Chapter 3.

Overview of chapter

This chapter proceeds in Section 2.1 with an informal examination of the casting industry, briefly reviewing its six thousand year history. Following this, in Section 2.2, methoding and the role of the Methods Engineer to ensure the production of sound castings is discussed. Casting is a highly complex process, and there is a strong need for computational assistance in an effort to empower Methods Engineers for yielding quality castings. This is discussed in detail in Section 2.3. Section 2.4 examines existing approaches for modelling the expertise of the Methods Engineer, and these approaches have been rule-based systems usually coupled with design-by-feature. A CBR approach for shape retrieval is finally proposed.

2.1 Introduction to Casting

Casting is concerned with the creation of metal objects made by pouring molten metal into a hollow cavity called the mould and letting the metal solidify in the hollow (Sylvia 1972). The casting process provides the ability to mass-produce a wide range of complex shapes and continues to be the most preferred process for the creation of complex metal designs (Ravi, Creese and Ramesh 1998). Indeed, the casting process is the most economic route for the production of geometrically complex metallic components of diverse sizes (Ravi and Srinivasan 1989). In the U.S. alone, 90% of all manufactured goods and capital equipment use castings as engineered components or rely on castings for their manufacture, a market that is worth on average 20 billion dollars, with over 14 million tons of casting shipped annually (American Foundrymen’s Society website 2000). In the UK, despite a downturn in manufacturing, the casting industry is still a major sector with annual sales of £2 billion and employing over 50,000 people (British Metal Casting Association).

Casting was one of the numerous breakthroughs that steered civilisation to new levels of advancement, empowering early man with a means for developing metal cutting tools, weapons, and utensils.

Approximately six thousand years ago in Mesopotamia, an area that is now situated in modern Iraq and Eastern Syria, foundry practice began (Simpson 1969). Copper was melted there using a forge fire and poured into stone moulds. This new discovery moved eastward into the Orient where it developed to an advanced level with the Chinese mastering iron casting in 600 B.C. and inventing the lost wax process. Both these processes were re-discovered centuries later, a recurring theme in the formative foundry industry, since early casting pioneers did not efficiently document their discoveries, instead relying on unwritten repositories of knowledge passed from master to novice. Around 1000 A.D. developments in the Orient ceased, but knowledge of casting turned westward into the Near East, the Mediterranean basin and the rest of Europe. However, it was not until 1000 A.D. that European achievements began.

Although the period of the 'Dark Ages' resulted in foundry developments being shrouded in mystery, it did not become a *lost* art. Indeed, the earliest Christian schools taught metal work, but carefully guarded their technical secrets. This resulted in the metallurgical field becoming the private property of the monks and clerics. The monk Theophilus in the 11th Century realised the importance of keeping manuals of foundry practice, but aside from his work there is a blank in the historical developments until the 15th century, when Vannoccio Biringuccio, the 'father of the foundry industry', collected the practical knowledge of foundrymen and published it.

Following the Renaissance, trade and commerce flourished, leading to the establishment of craft guilds. The guilds took complete control over foundry operations in the same way the church had. Although the guilds eventually deteriorated due to their monolithic status, they made the important rediscovery of casting iron, a metal that was formerly considered as 'corrupted'. Iron, with its low production costs affected the whole industry, and had a revolutionary impact on society. Yet, this had a detrimental impact on the environment when great forests were destroyed to stoke the blast furnaces; a problem not solved until 1730, when the use of coke as a fuel was started.

Although the world was becoming highly mechanised in the 19th century, casting was still a black art. With the collapse of the Tay Bridge in 1879, resulting from other faults excessively high porosity in its iron columns, there were increasing demands for industrial components of high strength and complexity. This disaster gave impetus for the scientific study of foundry processes.

Chvorinov, in 1940, answered the question: *How long does it take for a casting or part of a casting to solidify?* (Sirilertworakul, Webster, Dean 1993b). He put forward his now famous rule for solidification that predicted that the approximate freezing time of metal depended on its volume to surface area ratio, called the modulus (Ruddle 1971). Wlodower (1966) then built on the notion of modulus as a solidification parameter as an approach to foundry design. In this research, Wlodower's technique is used for a computer-based approach to shape representation and this is discussed further in Chapter 4. As Ravi and Creese (1996) elucidate, casting is a knowledge-intensive process and since the 1970s, the immense growth in computing power and cost reductions in hardware (Higginbotham 1995) have initiated developments in applying computational techniques to assist Methods Engineers using advanced solidification packages and applied AI, such as expert knowledge-based systems (Nanda, Smith, Haberle, Voller 1994). Software for casting design is examined in Section 2.3 of this chapter.

There is a range of casting processes (Beeley and Smart 1995). Sand casting is of special interest in this research, and castings using this process are created by packing a special sand that hardens rapidly like cement around a physical pattern (which is commonly made from wood) to form a mould. The pattern is pressed into the top (cope) and bottom (drag) of a mould box, thus leaving a hollow impression in the hardened sand. Then the mould is weighted down against the pressure of hot air and steam, and molten metal is poured to fill the mould, solidifying to produce a metal casting.

2.2 Methods Engineering

Methods Engineering is a crucial job in the foundry industry and is concerned with the design of the casting process, which involves filling the mould with molten metal and the ensuing solidification of the metal. This research is focused on automating an essential process of methoding, that is the automation of the human processes that allow the recalling of prior casting designs similar to a target design. This recollection of previous designs (cases) is a fundamental procedure in methoding, a skill perfected over many years. In essence, Wright (1992) has described the role of the Methods Engineer as:

> *"The function in the foundry context that ensures the casting supplied to the customer is of the quality he requires and that it has been produced within the costs previously estimated."*

Wright gives some of the functions that are involved in ensuring a good casting as:

- Cost estimating
- Scrap forecasts
- Orientation of the shape in the mould
- Position and sizing of feeder reservoirs
- Chill location

Acceptance criteria for casting quality are quantified against a particular component specification's 'fitness for its intended purpose', followed by non-destructive testing where possible internal discontinuities are detected (BCIRA 1992b):

1. <u>Surface inspection</u>: visual/tactile, magnetic particle, liquid penetrant.
2. <u>Internal inspection</u>: radiography, ultrasonics.

Initially, before any molten metal is poured into a mould, an engineering drawing that represents the design of the component is submitted to the Methods Engineer with a request to 'method' the shape for production by the casting route. The process usually involves a dialogue with the client, where possible re-design of the shape for a more efficient casting may be suggested.

The Methods Engineer must consider a variety of problems associated with how the metal solidifies in the mould. The most important consideration is the shrinkage, or volumetric contraction (Prasad and Kondic 1994) which accompanies solidification. Since metal in the mould freezes first at the boundaries, there is a possibility that isolated pockets of molten metal may form during freezing. Subsequent shrinkage of these pockets give rise to porosity and other casting defects. Therefore, the rate of metal cooling has a direct influence on the final microstructure and soundness (quality) of a casting. Moulds must be defined to feed metal to the casting so to keep it full as solidification proceeds.

Solidification of castings is a non-linear transient phenomenon (Ravi and Srinivasan 1996), and there are three stages of contraction in volume when the metal cools from liquid to being a solid, as shown for nickel-aluminium-bronze castings in Figure 2.1 (MOD 1979:3). The X-axis shows the temperature of molten metal as it freezes from 1200 C° to 0 degrees C°. The Y-axis shows the metal shrinkage in the mould cavity volume as the temperature of the molten metal drops. This figure shows a mould filled with molten metal at 1200°C. As the metal freezes, there is a contraction in the solidifying metal as the temperature drops to room temperature. The three stages the molten metal passes through are:

- Liquid Contraction: Occurs as the liquid metal cools from the pouring temperature (1200°C) to liquidus temperature (1070°C). There is approximately a 4% reduction in the volume of the liquid metal.

- Solidification Contraction: Takes place as the metal cools from being completely liquid at a liquidus temperature (1070°C) to being completely solid at solidus temperature (1050°C). There is a sharp reduction in volume (approximately 4%) as the temperature of the metal falls by just 20°C.

- Solid Contraction: Occurs when the solid casting contracts from the mould walls as the temperature drops from the solidus temperature to room temperature. This contraction does not affect the soundness of a casting. However, the Methods

Engineer must ensure that the pattern dimensions are larger by a 2.3% than that of the final casting. This allowance is known as the *Pattern Makers Contraction Allowance* (MOD 1979:2).

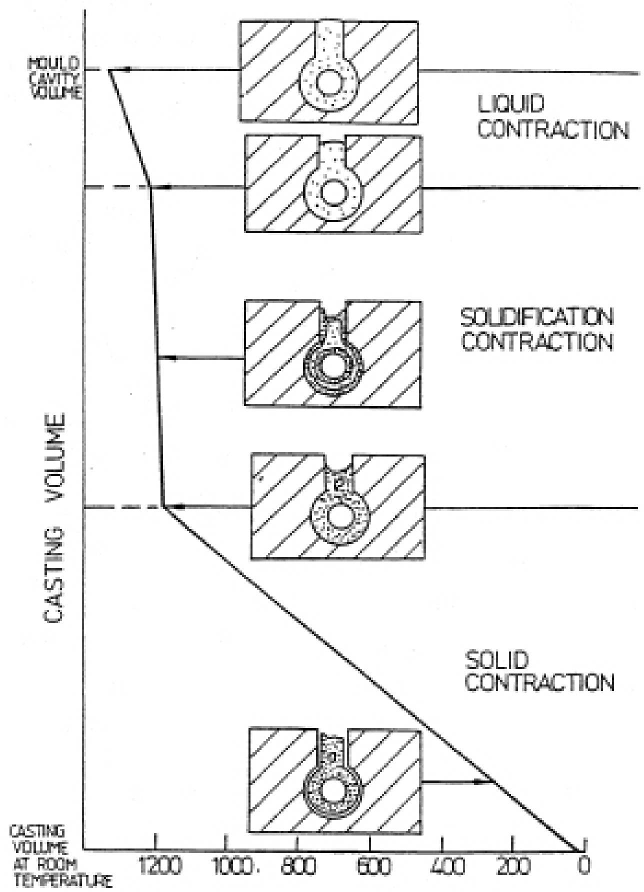


Figure 2.1 Solidification of Nickel-Aluminium-Bronze in a Mould (adapted from MOD 1979:3)

The Methods Engineer places *feeders* and *chills* at strategic places in the mould to ensure that no such isolated pockets of molten metal can form at any stage. Feeders supply liquid alloy into the casting section to account for the volumetric contraction from liquid to solid, and chills are used to hasten the freezing process. Although not every casting needs chilling, feeders are necessary to ensure directional solidification, so that molten metal solidifies in the feeder. By using design knowledge, feeders can be minimised by using the skilful placement of chills, and this can lessen the scrap rate.

To make this clearer, Figure 2.2 (MOD 1979:21) shows a plate without a feeder, and the resulting porosity. Figure 2.3 (MOD 1979:21) shows the same plate with a feeder, positioned so that the whole of the casting solidifies *directionally* towards the feeder. Using a feeder, the shrinkage porosity is centralised in the feeder, *not* in the body of the plate as Figure 2.2 shows.

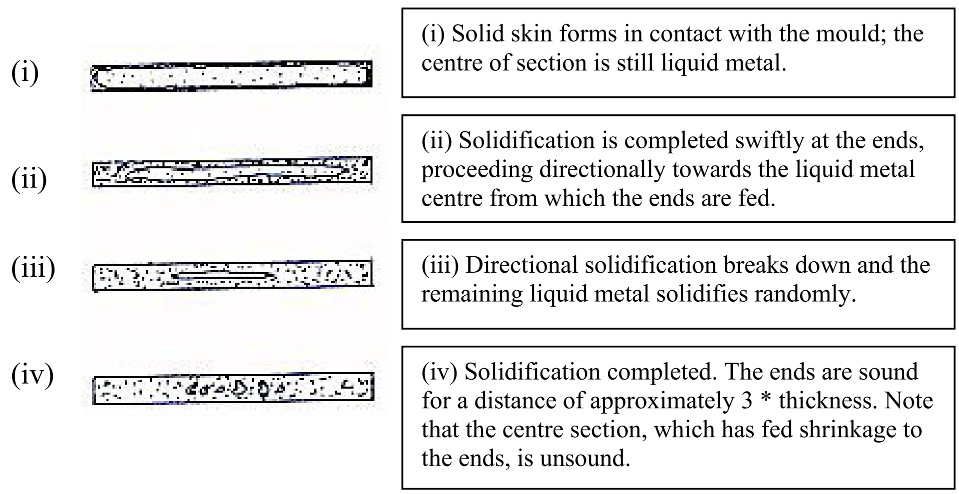
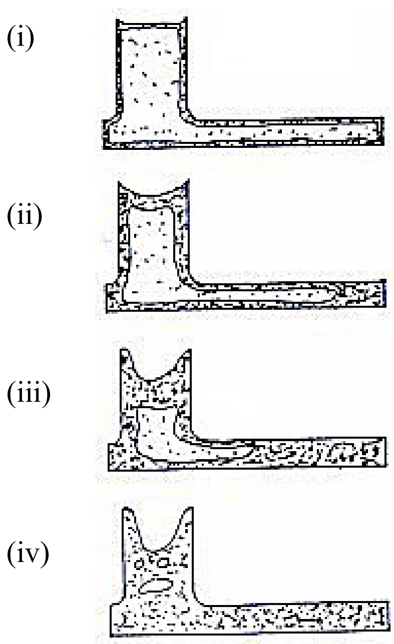


Figure 2.2 Solidification of a Plate without a Feeder (MOD 1979:21)



(i) Solid skin forms in contact with the mould.

(ii) Directional solidification begins from the ends.

(iii) The casting solidifies directionally towards the feeder if the maximum feeding distance is not exceeded.

(iv) The casting is sound, because the feeder has solidified last and all the shrinkage has been localised in the feeder, rather than in the casting itself.

Figure 2.3 Solidification of a Plate with a Feeder (MOD 1979:21)

There are other considerations in addition to placing feeders and chills, and these include the design of the running system (which fills the casting with molten metal) and the orientation of the shape. These are all crucial decisions that have an important effect on the casting, and the knowledge to do this is gained over many years of methoding. The key to successful methoding therefore, is to understand how metal solidifies in the mould and the associated problems as this occurs, applying knowledge to ensure a sound casting.

A simple model constructed by the Methods Engineer as they attempt to assess the solidification is the *modulus model* (Chvorinov 1940, in Wlodower 1966), which is a crude approximation of the dynamic cooling of a casting. The Chvorinov Rule was a significant advancement of this objective and postulated that the approximate freezing time of any casting is a direct function of the ratio of its volume to its surface area known as its modulus, *M*, and is expressed as:

$$M = K(V/A)^2$$

where *M* is the modulus, *K* is a constant dependent on the metal being cast and on the mould materials, *V* is the casting volume and *A* is the cooling surface area of the casting (Piwonka 1995). The modulus is discussed further in the following chapter, with emphasis on how it provides natural keys for shape retrieval.

2.3 Software for Casting Design

This section examines the role of software relevant for use in the casting industry that can empower the Methods Engineer. A taxonomy of casting software is presented and within this taxonomy the application of CBR as a mechanism for retrieving previous design cases is investigated, showing how the modulus model acts as a key to accomplish the task of shape retrieval.

2.3.1 The Need for Casting Software

The casting process is highly complex, and some 2000 variables can be involved in the production of a single casting (Sillen 1991). Indirect proof of this complexity is revealed in scrap rate figures for the industry, and these are in the five to seven percent range (Ravi, Creese and Ramesh 1998). The scrap rate is high, because usually several test pours have to be made before a new design is used for production runs (Sillen 1991); the wastage metal becomes ‘scrap’. Sillen goes on to say that subsequent design revisions are both costly and time consuming, primarily because product designers have insufficient knowledge about casting processes and have problems judging the effect that design features have on castability, in quality, costs and productivity. Orogo, Callihan, Sigworth and Kuhn (1993) reflected on this problem in their vision of computer-aided casting in the year 2000, stating that the casting process is hindered by human error decisions, which necessitate expensive redesign and remakes. They claim the reasons for this result from not fully understanding the inherent properties of solidifying metal in the mould. Ravi (1999) gives a gloomy summary that despite casting being many thousands of years old, the process still happens to be more of an art than a science.

The economic importance of sound methoding practice to the casting industry therefore is very high. For this reason, foundries are coming under heightening demands to improve quality and are increasingly turning to advanced software in an effort to diminish or expel the costly need for trial-and-error prototyping (Clegg 1986; Clifford 1992; Estrin 1994). In the UK, foundries make up an important sector of the UK's manufacturing industry and there are considerable benefits to be gained by improving methoding practice by lowering scrap rates and in the general improvement of casting quality. In 1987, an audit group commissioned by the Department of Trade and Industry (DTI) recommended a UK initiative in the field of computer-aided design for casting (Knight, Cowell and Preddy 1995). The majority of the research which has resulted from this initiative has been concerned with the progress of faster and more accurate numerical routines for the simulation of the complex physical processes of flow, freezing and stress involved in casting solidification, rather than AI approaches such as using CBR.

2.3.2 A Taxonomy for Casting Software

Casting software can be broadly classified in five ways (Jolly 1996) using the software taxonomy as shown in Table 2.1. This taxonomy is briefly explored below, showing examples of the types of software, both commercial and academic prototypes within each category.

Category	Name	Description
I	General	General-purpose codes for modelling heat transfer and shape mass. Non-casting specific.
II	Foundry FE Method	Finite Element based code aimed at the foundry industry, usually embodying functions for defect prediction
III	Foundry FD Method	Finite Difference based code aimed at the foundry industry, usually embodying functions for defect prediction
IV	Heuristic	Aimed specifically at the foundry industry. Broad sweep of codes, including knowledge-based, frequently with quasi FD calculations.
V	Mixed	Codes which have a rapid solidification, usually based on a Chvorinov type calculation, followed by a FDM/FEM in-depth analysis

Table 2.1 Taxonomy of Casting Software (Jolly 1996)

2.3.2.1 Category I - General

Software in this category is applicable to the modelling of heat transfer and fluid flow based on general-purpose finite element (FE) or finite difference (FD) codes. Ravi (1999) explains that these simulation programs in essence decompose the model into many thousands of elementary elements (bricks in FE and tetrahedrons in FD) and successively apply the heat transfer formulae equations for conduction, radiation and convection and the Navier-Stokes equations for fluid flow to all elements. The calculations are repeated many times until solidification is complete. The progress of filling or solidification in the mould can be visualised through colour coded plots.

2.3.2.2 Category II, III – Foundry FE/FD Method

Both FE and FD methods are used extensively in Category II and III codes, which differ from Category I codes in that the codes are specifically aimed at the casting industry and usually have some functions for predicting defects such as shrinkage cavities, porosity and air entrapment. They allow the Methods Engineer to be able to review various options for feeding and gating, and decide on an optimal set-up before any casting is put into production for mould filling.

FD methods codes have produced very good results with tolerable accuracy; for example MAGMASoft, SIMULOR (Rigaut, Meyer, Charbonnier, Bourg 1995), MAVIS and equally with FE methods codes; for example, ABAQUS, ADINA, ANSYS (Jolly 1996). More advanced numerical software using computational fluid dynamic techniques are under development (Cross 1993; Chow, Bailey, Cross and Pericleous 1995).

Ravi (1999) explains that FE programs are far more complex than FD programs to develop, although FE programs can model a casting more accurately than FD methods. However, only a few organisations can afford the costs of the software, as well as the work-hours required to input the complex geometries that typify today's design expectations. This is particularly the case with most foundries being classed as small manufacturers (Preddy, Knight, Cowell and Mileman 1997).

2.3.2.3 Category IV - Heuristic

Software within this category is termed "black box" and encompasses both knowledge-based and fast "look see" rapid mould filling solidification models based on Chvorinov's modulus calculations, such as SOLSTAR (Corbett 1989), NOVACAST and CADCast (Huang, Webster and Dean 1995a).

Recently, intelligent knowledge-based techniques have become increasingly prominent for helping to diagnose casting defects (Bradley, Adams, Gadh, Mirle 1993; Kluska-Nawarecka 1996) and to assist in the methoding of castings (Cowell, Knight and Preddy 1993; Knight, Cowell and Preddy 1995).

Much of the research and prototype software within this category has attempted to use geometrical feature extraction of a casting prior to fault analysis by rule-based expert systems and feature modelling. A critique of existing approaches used to model casting shapes is given in Chapter 3 – Knowledge Elicitation. Further research has focused on key areas of methoding, and these can be split into four research categories, as shown below; referencing key research papers within each section:

A. **Feeding and Gating**: (Upadhya, Paul and Hill 1993; Darwish 1995; Zhang, Webster and Dean 1995).

B. **Defect Analysis**: (Natarajan, Chu, Kashyap 1989; Sillen 1991;Webster, Weller, Sfantsikopoulos and Tsoukalas 1993; Webster 1995).

C. **Manufacturing Evaluation**: (You, Chu and Kashyap 1989; Nanda, Smith, Voller and Haberle 1995; Ravi 1996a).

D. **General**:

- *Determining the technological procedure for manufacturing castings* (Stoiljkovic, Mitrovic, Stoimenov and Milovanoic 1994; Nanda *et al* 1994).
- *Alloy selection* (Sirilertworakul, Webster and Dean 1993c).
- *Process planning systems* (Nealon and Firth 1989; Ravi 1996b).
- *Finite Element Mesh design* (Dolsak, Jezernik and Bratko 1994; Nagasawa, Miyata, Murayama and Sakuta 1996).
- *Cost estimation, parting generation and analysis, core identification, casting inspection, casting information management*; see the review paper by Ravi and Creese (1996).

Ransing, Srinivasan and Lewis (1995) have pointed out that the casting process is an ideal candidate for expert system utilisation. Indeed, Creese and Waibogha (1987) were among the first researchers to suggest casting defects could be reduced or eliminated by utilising a methodology predominately based on expert system technology. Some of the reasons why knowledge-based technology is seen as the key for intelligent casting software is discussed by Darwish and El-Tamimi (1996) who base their argument on the claim that no single Methods Engineer can practically be expected to know or remember all aspects of casting design and previous casting designs.

There are, however major problems with the knowledge-based approach (Kolodner 1993; Watson 1994) in that expert knowledge is difficult to elicit and codify. There is now a rapidly growing interest in using CBR techniques for engineering (Althoff, Auriol, Barletta, Manugo 1995) particularly in the computer-aided design and manufacturing field, and in the engineering community. The role of CBR is examined later in Section 2.4.2 of this chapter.

Additional software, both commercial and academic prototypes within Category IV are in existence, and it is beyond the scope of this thesis to investigate each of them in detail. However, as Ravi and Creese (1996) point out, the software available supports only a limited range of tasks that the Methods Engineer carries out and most of the software packages are both expensive and need skilled technical support to operate. Furthermore, the programs cannot be interfaced easily since each one is bespoke. For these reasons, the Casting 2000 Project was proposed by Ravi (1996a) with the aim of creating an integrated family of intelligent software packages that can assist the Methods Engineer at each stage of the methoding process.

2.3.2.4 Category V - Mixed

The codes in this final category of mixed software usually have a rapid front-end solidification model, again based on Chvorinov's Modulus calculations, and for more detailed analysis a FD methods module. Some available packages, such as *CADCast* (Sirilertworakul, Webster, Dean 1993a; Huang, Webster and Dean 1995b) combine both a solid modeller and a knowledge-base (used for choosing an alloy and a casting process (such as sand casting) relevant to a particular component specification).

2.3.3 Casting Industry Software Requirements

Jolly (1996) in his review of numerical analysis packages found that the software development progress is rapid and subject to further evolution. In his conclusions he commented that the foundry industry is looking for software that can not only predict problems that happen during metal solidification (such as shrinkage porosity) but also, having predicted these problems, to propose intelligent solutions to the problems found. Currently, no single package can achieve this. Thus, Jolly proposes using a hybrid approach that combines experiential data with a numerical analysis approach.

A chart showing the distribution of packages is shown in Figure 2.3. Although Jolly did not provide a cost-benefit analysis, Ravi (1999) reviewed the results of eleven casting simulation packages in the USA used by 154 foundries and found that the software reduced labour costs by 40% and improved casting yields by 25%, showing that simulation of the casting processes, prior to mould filling can reduce casting defects. Ravi, Creese, and Ramesh (1998) investigate the growth rate of casting software, and found that among 33,500 foundries worldwide, 1000 used simulation software, and that the number of foundries using such software is increasing.

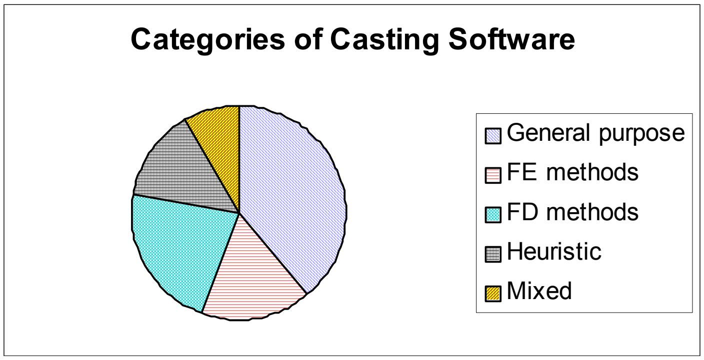


Figure 2.3 Summary of Casting Software; adapted from Jolly (1996)

2.3.4 The Computer-Assisted Methods Engineering Process

The Methods Engineer usually employs a number of computer-based tools to establish essential design parameters and to predict the likely outcome of the physical process of filling and casting. For the initial stages of methoding these tools need to be fast and easy to use: simple models based on the cooling modulus or fast mould-filling models. However, the most important ingredients are the Methods Engineers' practical experience and their use of these models. Figure 2.4 shows the Computer-Assisted Methods Engineering process.

The process is an iterative one, in which tentative designs are evaluated using these tools. Initially, there is usually a dialogue with the client, where possible re-design of the shape for a more efficient casting may be suggested. The Methods Engineer then decides on the orientation of the mould during filling, then designs the positions and sizes of feeders and chills. For this task, the engineer relies mostly on experience with hand calculations of the modulus of elements of the shape to gain an approximation of the solidification time taken for the element to cool. The design can be evaluated against various simulation models, and changes are made until the simulations are satisfactory. As Natarajan *et al* (1989) point out, the casting process is based heavily on Methods Engineering experience, and the design of a casting is an iterative task between casting designers and Methods Engineers.

Several software tools may be used to assist the methoding process as delineated in the software taxonomy in Section 2.3.2. Among these, are CRUSADER, FEEDERCALC and SOLSTAR (Corbett 1989), which support the preliminary design stages, and slower, more detailed models such as MAGMAsoft and SIMULOR, which support the simulation stages. CRUSADER and FEEDERCALC give numerical support on such aspects as feeder sizes and feeder distances, but do not attempt to give experiential advice; for example, re-design advice or mould orientation. SOLSTAR is often used as a fast solidification model, which can check a given design, or give information on feeder positioning.

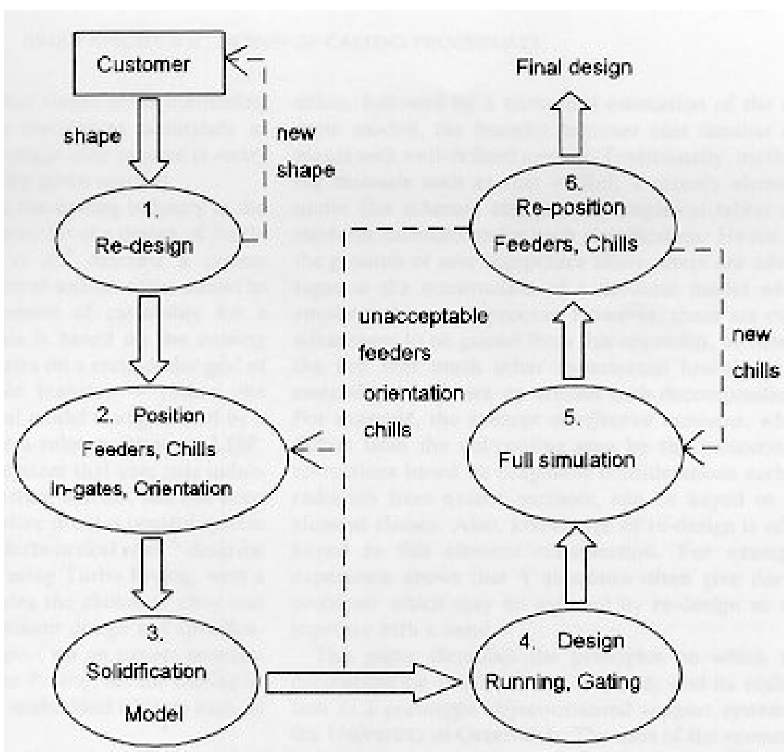


Figure 2.4 The Computer-Aided Methods Engineering Process

2.4 Approaches for Modelling the Expertise of the Methods Engineer

Three general approaches have been used in casting research for modelling the expertise of the Methods Engineer. The first and most widely used is traditional numerical modelling software to predict how a casting solidifies in the mould; although this does not include intelligent feedback on design. Other ways have predominately used rule-based expert systems, typically coupled with *design-by-feature* to drive geometric reasoning. In the past few years though, CBR has become a more viable contender in engineering, yet there have been only two systems currently in use in the foundry industry. Table 2.2 shows the category of techniques used for modelling the expertise of Methods Engineers.

1. Traditional Numerical Modelling	
2. AI Techniques:	2.a Rule-Based Expert Systems
	2.b Case-based reasoning (CBR)

Table 2.2 Approaches Used for Modelling the Expertise of Methods Engineers

2.4.1 Rule-based Expert Systems

The role of Expert Systems was discussed briefly in Section 2.3.2 - A Taxonomy for Casting Software. Expert systems constructed using rules have been the mainstay in predominant applications of AI (Marir and Watson 1995), in which knowledge is encoded with *IF...THEN* rules, and an inference engine directs the application of rules during the problem-solving process (Hedberg 1993). Much of the research for intelligent casting design has attempted to use geometrical feature extraction of a casting prior to fault analysis by rule-based expert systems.

Features have been primarily classified as protrusions (such as boss, rib) and depression type (such as hole, slot, pocket), relevant to the application area (Ravi 1996a). Ravi goes on to discuss that there have been relatively few research projects aimed at identifying the features in the casting domain. He identifies some of the features that could be employed for modelling purposes, including:

- Global and local symmetry
- Parting line
- Largest cross-section
- Bosses
- Fillets
- Solidification modulus
- 3-D corners
- Projections
- Depressions

Feature recognition algorithms are predominately rule-based. An example rule is given in Table 2.5, from the system EXCAST (You, Chu and Kashap 1989). This rule (Rule-50) checks whether a particular section thickness causes unsoundnesss.

Rule-50:
IF Primary feature is 1 **AND** Direction of hole is left OR right **AND** The distance from bottom of hole to side is less than minimum economical section thickness **THEN** Give illegal dimension message and suggestion for economical section thickness

Figure 2.5 Rule for Determining Unsoundness in a Particular Section Thickness

Other feature-based research in casting has been carried out by Luby, Dixon and Simmons (1988) who have designed a prototype system called *Casper*, which defines a shape grammar, allowing the creation of designs by the use of a vocabulary of familiar geometric features. The design is then evaluated for manufacturability by the construction of the modulus model from the features. Woodward and Corbett (1990) have taken a similar approach, concentrating on design rules for aluminium alloy die-casting.

Chung, Patel and Cook (1990) describe two applications for feature-based modelling combined with geometric reasoning, one application area being a 'critic' for predicting potential defects in gating designs for investment casting[4], which proposes design alterations to the Methods Engineer. Hill and Berry (1991) have carried out similar research for the automatic feature extraction from the boundary surface representation of a solid CAD model. Various geometrical information, for example, the thickness of sections can be extracted and utilised by rigging design rules. Three of the design rules the researchers have formulated for positioning feeders are shown below:

- *Feeders are located near thick sections*
- *Minimum distance between feeders should be maintained*
- *If blind feeders[5] are used, height:diameter ratio of 1:1 to 3:1 should be maintained*

Currently, no new tools of practical importance have yet arisen from this research for casting design. A key reason for this is the fundamental problem of rule-based systems concerning their operation in domains that are not well understood, resulting

[4] A process in which a mould is produced by surrounding an expendable pattern with a refractory slurry that sets at room temperature. After this, the wax or plastic pattern is removed using heat prior to filling the mould with molten metal. When a wax mould is used, the process is called the *lost wax process*.

[5] Blind feeders are a special type of feeder that work on the principle of using a 'fire cracker' core to puncture a hole in the steel shell that sets up as the blind feeder solidifies (see Rowe 1991).

in systems that tend to be brittle and hard to maintain (Mott 1993). Indeed, there is currently no comprehensive model of the foundry process because of the lack of fundamental knowledge (Phelps, Heine and Uicker 1989). Furthermore, these prior approaches have been unrelated to the experience of human foundry experts which is based on a natural shape decomposition scheme linked with prior knowledge of cast shapes.

2.4.2 Case-Based Reasoning (CBR)

CBR is based on the premise that humans often solve current problems by association or analogy (Kolodner 1992; Hedberg 1993; Marir and Watson 1994; Aha 1994, 1998; Lenz, Burkhard, Bartsch-Sporl, and Wess 1998) and has been described as one of the success stories of AI research (Watson 1995). The conjecture behind CBR is that when faced with a new problem, humans often remember a previous experience and adapt it to suit the new problem they are faced with. The process is relevant in this research, in that Methods Engineers solve new methoding problems by recalling earlier methoded solutions. Leake has succinctly called the CBR process as one of *reasoning by remembering*.

Riesbeck and Schank (in Watson 1995:4) define CBR as follows:

"A case-based reasoner solves new problems by adapting solutions that were used to solve old problems."

CBR derives initially from the philosophical investigations of Wittgenstein (1953) who challenged the classical Aristotelian view of learning concepts; for example, a "grandmother" is the mother of a parent. Instead, Wittgenstein reasoned that categories (such as games) are characterised by 'family resemblances' between members of a family, and this led on to the work of Rosch (cited in Pinker 1999) in 1970 who introduced many of Wittgenstein's ideas into psychology. Later, Schank researched dynamic memory (Schank 1982, cited in Marir and Watson 1994), and hypothesised that humans use 'scripts' when they face a situation they have not met before. When faced with a new situation they recall a previous script, adapting it to suit the present circumstances. The usage of cases that hold antecedent experience evolved from this concept. Two central elements of a CBR system are thus focused on case retrieval (recalling cases that match the present situation) and adaptation (reforming the case to the needs of the current problem). A case-library in a CBR system models human memory by holding cases that represent a repository of experience.

CBR can be described as cyclical framework using four processes called the four RE's (Aadmodt and Plaza, 1994; Holt and Benwell 1996): retrieve, reuse, revise and retain. The process is illustrated in Figure 2.6. In summary, the cycle starts with a new problem (known as a *case*) presented to the CBR system. The CBR system *retrieves* the most similar case(s) which are solutions to the problem case. The cases are stored in what is known as a case base (a database of cases). The information and knowledge in the retrieved case(s) are *reused* in an attempt to solve the new problem. The solution may need to be *revised* (modified*)* before reuse. Finally, the new solution is *retained* by storing it in the existing case base. The CBR cycle is examined further in Chapter 6 - Section 6.1 with emphasis on how it is applied in this research.

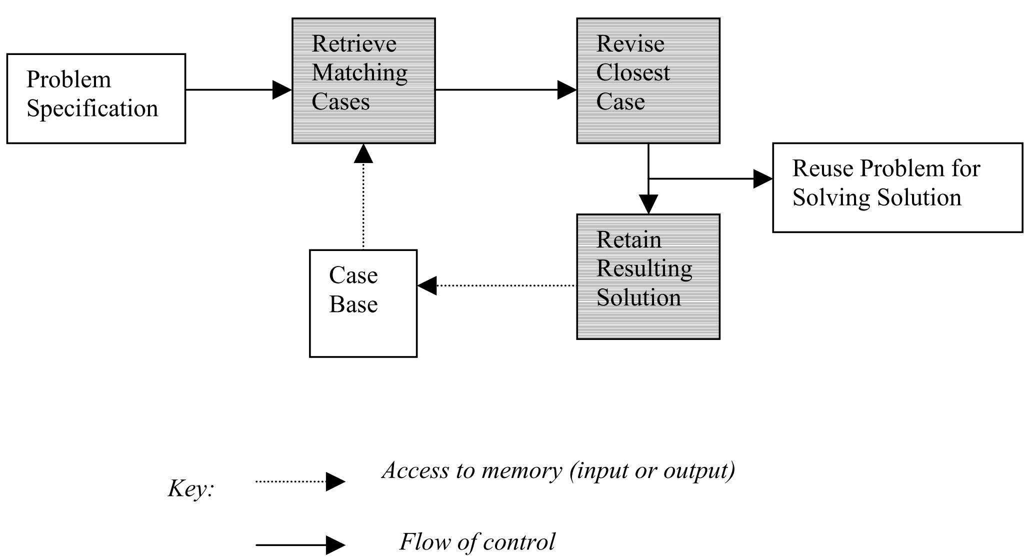


Figure 2.6. The Case-based Reasoning Cycle (adapted from Maher 1996)

No deep domain model is required in CBR (Marir and Watson 1995), whereas using a rule-based approach entails a good understanding of the domain model so that the underlying rules can be developed. In the casting domain, the expert for much of the time does not know why they would choose a particular course of action; their knowledge has become *intuition*. Thus, acquiring deep knowledge of casting in rule format is difficult. This is echoed by Taylor (1997) in that many problem domains are rich in judgmental or intuitive experiential knowledge, and because of this it is difficult to encode this knowledge directly in the form of knowledge-based production rules; this is where CBR can be of most practical use (Hennessy and Hinkle 1992). Marir and Watson (1995:108) note that:

> *"The KBS community was seduced by rules and neglected the truism that experts solve problems by applying their experience, whilst only novices attempt to solve problems by applying rules they have recently acquired."*

Bradley and Gupta (1995) in their classification framework for CBR systems argue that one of the main strengths of CBR is that is a powerful approach to solving problems that demand experience, intuition and judgement. Kolodner lists several advantages that CBR offers over rule-based reasoning, and these are relevant in the context of Methods Engineering. Kolodner lists these advantages as follows:

- CBR proposes solutions to problems quickly, avoiding the time necessary to derive those answers from scratch.
- CBR proposes solutions in domains that the human does not understand completely.
- CBR gives a means of evaluating solutions when no algorithmic method is available for evaluation.
- Cases are particularly useful in interpreting open-ended and ill-defined concepts.
- Remembering previous experiences is useful for avoiding future problems.

- Cases help a reasoner to focus on important parts of a problem, by indicating important problem features.

It is generally agreed in the AI community that CBR comes closer to the human decision making process than the long established expert system model of reasoning (Burkhard, Kuhnel and Pirk 1994). Instead of inferring a solution from first principal rules, previous analogous problems are replayed in the expert's mind (Ketler 1993; cited in Taylor, 1997). Kolodner (1993) remarks that:

"If we watch the way people around us solve problems, we are likely to observe case-based reasoning in use all around us."

This claim is also supported by findings from cognitive psychological research (Aadmodt and Plaza 1994). Kolodner (1993) adds:

"As a method for building intelligent reasoning systems, case-based reasoning has appeal because it seems relatively simple and natural. While it is hard to get experts to tell you all the knowledge they use to solve problems, it is easy to get them to recount their war stories."

Mott (1993, in Cowell, Knight and Preddy 1994) points out that CBR fills the gap between knowledge-intensive technologies such as mathematical modelling and rule and frame-based systems on the one hand, and knowledge-limited technologies such as neural networks and pattern recognition systems on the other. The former approach is natural where expertise is easy to codify using algorithms, rules or semantic networks, the latter approach is natural in domains where expertise is either non-existent or very thin on the ground, as in image processing or the prediction of currency fluctuations.

2.4.3 Applying CBR for Casting Design

The value of using CBR for expressing and applying geometric design knowledge for design is growing rapidly within the engineering research community in general (Hua, Faltings and Smith 1996; Bilgic and Fox 1996), primarily as a result that the quality of designs is predominantly determined by its geometry. In the casting industry the value of design knowledge is being widely recognised. Nevertheless the management of design knowledge is often *ad hoc* in some respects, and there are many advantages of a common casting design database that could utilise CBR techniques. Design histories are often lost or consigned to unsearchable paper files. Methods Engineers retire or move, leaving inadequate design records (Price, Peglar, and Bell 1993). Information is also not traded between companies, so that a supplier must start from scratch without benefiting from previous designs.

Although there has been much research in applying feature-based technology in the domain of structural and architectural design (Alberts, Wognum and Mars 1992;Yeh 1997), there has been a paucity of research in applying CBR in the foundry industry, apart from the work of Price, Pegler, Ratcliffe and McManus (1997) in a CBR system for solving problems during the manufacturing of aluminium components, and *Wayland* (Price and Pegler 1995), a CBR system for determining the setting of parameter values on an aluminium die-casting machine, which operated on feature-value pairs. Feature-based approaches offer a powerful representation scheme and clearly point the way to automating the retrieval of cases by CBR. As Duffy points out (1997:71):

> *"Design experience represents one of the most powerful resources a designer possesses."*

However, the advantages of coupling design-by-feature with CBR as a way for retrieving casting shapes has not yet been fully realised by earlier researchers. Scherer, Berkel, Schlageter and Schultheiss (1993) in research that investigated the use of neural networks for reuse of existing design objects, summaries some of the improvements obtained by applying CBR in design as:

- **Systematic reuse of already existing knowledge:** Design knowledge that already exists can be recycled instead of having to be rediscovered. The researchers succinctly put this point across as - ‘Do not discover the wheel twice’.

- **Time:** The time for optimising a CAD/CAM object can be minimised. From a foundry domain perspective a historical database of methoded casting designs means that new problems can be matched quickly by sophisticated CBR algorithms with existing designs, and this may significantly reduce the amount of iterations in methoding a new design.

- **Quality:** Quality standards can be improved by matching new designs with designs that represent a certain level of quality, both good and bad.

2.5 Conclusions

From the examination in this chapter it is evident that casting is an important manufacturing process. As pointed out, in the US alone, 90% of all manufactured goods use castings and the UK annual turnover is two billion pounds, employing over 50,000 people.

Although the foundry industry has a rich and exciting six thousand year history, it does not follow that everything is precisely known to make fault-free castings. On the contrary, casting is still more or less an art rather than a science and it has only been in the last sixty years that mathematical procedures (such as Chvorinov’s modulus concept) and computational assistance in the form of solidification simulation codes have been applied to assist the foundry industry in its goal of improving quality and lowering the five to seven percent scrap rates.

Since knowledge of how to cast a product is essentially knowledge-based, computer programs have been developed that try to encode human knowledge of casting. Foremost among these have been intelligent knowledge-based systems (IKBS), using rule-based expert systems; these have achieved only limited success. More promising has been the application of CBR systems, an approach that is psychologically closer to the way Methods Engineers recall previous similar casting designs when methoding a new casting design.

The modulus model, as formulated by Wlodower, and how Methods Engineers mentally decompose a shape into various sub-elements to gain a simplified model of solidification was investigated. This model provides natural keys for a shape retrieval system, based on the CBR archetype. The following chapter builds on these ideas and examines the process of shape componentisation used by Methods Engineers, the starting point in providing a sound platform for subsequent shape retrieval.

Chapter 3 Knowledge Elicitation

This chapter considers the knowledge of the Methods Engineer and how knowledge of 'shape' is crucial for ensuring quality castings. The knowledge elicitation focused the investigation on an existing technique using componentisation of 2-D section 'slices' into elementary components. The first part of this chapter investigates the importance of shape in design problems, the primary problem being porosity. Following this, the significance of casting geometry when it comes to ascertaining the solidification time of molten metal using the modulus concept is examined. Given an arbitrary casting design, techniques must be found to predict its casting solidification time. In the background on Methods Engineering in Chapter 2, the discovery of the Chvorinov Rule was a step in this direction, giving a way to componentise shapes into elementary components. In the early stages of the research, it became apparent that the Methods Engineer considers 3-D shapes as if the shape were made up of 2-D sections, working from engineering drawings that represent various cross sections of horizontal and vertical sections. Examples are given in this chapter showing the use of cross sections in methoding a shape. There is also a section on alternative approaches to shape representation, which have been based on feature extraction from solid modelling techniques.

Overview of Chapter

Section 3.1 examines the importance of shape in ensuring quality castings, explaining how Methods Engineering knowledge is crucially linked to shape for ensuring a quality, or *sound* casting. A prime concern of Methods Engineering is being able to predict the solidification time of a shape. Section 3.2 gives evidence for a natural shape decomponentisation, the 'modulus' ratio, which is employed for this purpose. Section 3.3 discusses existing approaches to modelling shapes based on feature extraction from solid modelling constructions. Section 3.4 explains that Methods Engineers work from 2-D cross-sections, and that much of their knowledge is linked to this component classification of plane sections.

3.1 The Importance of Shape for Casting Design

Shape, or the geometry of a casting, is of prime concern in casting design. For instance Beeley (1972:323) states:

> *"The detailed shape of a casting is important from the point of view both of engineering function and suitability for the casting process."*

The main concern of the Methods Engineer is to consider shape features, this having an important effect on casting manufacture; for example, a shape with thin sections can cause unsoundness, and problems involving junctions are a major cause of porosity. Within the foundry industry it is considered bad casting practice to use an initial engineering drawing without first considering the implications of possible design modifications that could help to ensure internal soundness. Knowledge to ensure soundness is:

"derived from previous experience with castings of similar shape."

(Beeley 1972:325)

Once the Methods Engineer has an engineering drawing at their disposal, the design is evaluated using his own knowledge and experience of casting practice. Changes in casting design can be made in various ways to ensure soundness, and the main considerations are related to problems concerning porosity: section thickness, pouring and fusion problems, and the location of feeders and chills to ensure directional solidification. These problems are the main determiners of porosity. There are numerous benefits to be gained from correcting casting problems prior to moulding (Niyama, Uchida, Morikawa, Saito 1982) and these include a reduction of materials, energy and time, all of which lower the production costs.

This research concentrates on the domain of porosity problems, and these are considered as being among the most significant problems that can occur during metal solidification (Beeley 1972; Campbell 1991). Porosity problems are used to validate the similarity metrics (see Chapter 5), which is the main purpose of this research. Evaluation of these metrics are dealt with in Chapter 6.

There is a huge amount of published casting design knowledge available to practising Methods Engineers, such as the 'Casting Design Handbook' (American Society of Metals 1962). Clearly, it is beyond the scope of this thesis for a detailed examination of all the various casting problems and solutions, since these are well discussed in the casting literature. Instead, the dominant areas where problems occur and how casting design knowledge can be applied for re-design to secure soundness is briefly outlined.

These problem areas have been divided into three problem domains:

1. Porosity
2. Section thickness
3. Junctions

3.1.1 Porosity

As discussed in Chapter 2 - Section 2.2 on the background of the Methods Engineer, the soundness of a casting is crucially dependent on the relative course of solidification through its shape (Beeley 1972). As the metal freezes in a mould, shrinkage occurs, because metals have a greater density in the solid state than in the liquid state (Sirilertworakul, Webster and Dean 1993). The reduction of volume that occurs during solidification can result in the formation of damaging cavities, known in the industry as *porosity* (Campbell 1991).

Castings are designed for directional solidification, which means that the molten metal freezes progressively towards reservoirs of metal, called *feeders*, from sections with low moduli to those of higher moduli (Zhang, Webster and Dean 1994). Feeders act as reservoirs of molten metal to counteract the shrinkage of metal as it turns from liquid to solid. In an attempt to eliminate porosity, feeders and chills are placed at crucial places in the mould.

However, every effort must be made to reduce the use of feeders (Casting Design Handbook 1962; Beeley 1972) because over-use can lower the yield of metal poured and contribute to the scrap rate.

An essential parameter in achieving lower yields is the *feeding distance*. This is the distance that liquid metal can pass along a parallel section of a casting before the casting exhibits porosity (Campbell 1991; Zhang *et al* 1994). The feeding distance rule for plates is (Campbell 1991):

$$L_d = 4.5T$$

where T = thickness. Of this total distance, $2.0T$ results from the feeder and the remaining $2.5T$ is a consequence of the chilling effect that comes from the edge of the casting. This means that a parallel section can be fed with molten metal 4.5 times its thickness before porosity occurs.

Feeding distances give Methods Engineers a way to calculate the spacing of feeders, enabling the number of feeders to be reduced to the smallest number possible within the boundaries of acceptable ranges. Using fewer feeders can help reduce the scrap rate. Figure 3.1 shows an example of the feeding distance relationships for plates (Campbell 1991), displaying how the addition of a chill extends the feeding distance. Chills extend the feeding distance by 50mm; therefore long parallel sections can be fed without using extra feeders. In the bottom casting of Figure 3.1, the feeding distance can be more than doubled; that is, by $9T$+100mm.

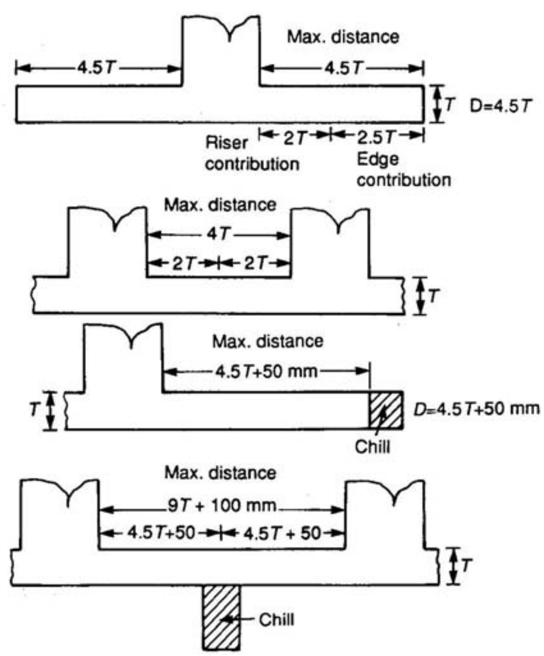


Figure 3.1 Feeding Distance Relationships for Steel Plates (Campbell 1991:188)

3.1.2 Section Thickness

The weight of a casting can be reduced by using thin sections and therefore can promote a more favourable strength-to-weight ratio. BCIRA Cast Metals Development Centre (BCIRA 1992a) point out that using excess metal can have a serious effect on total costs, in terms of melting costs, and the removing and recycling of scrap metal. Costs also sharply increase especially when a costly alloy is to be used. However, a uniform section thickness less than twelve millimetres are problematic, because it is difficult to ensure directional solidification of such uniformly thin sections towards an adequate source of feed metal, and as a result of this shrinkage porosity occurs (Beeley 1972). Figure 3.2 shows an example of an aluminium sand casting for an aircraft structural application[6], and this casting suffered from shrinkage along a 3/32 inch (≈ 2 mm) wall. Although the chilled bottom flange filled adequately, the castings were rejected because of micro-porosity and shrinkage in the thin wall section. Increasing the wall thickness to 5/32 inches (≈ 4 mm) eliminated this shrinkage. As the lower figure shows, when the wall thickness was increased, this provided conditions for directional solidification, resulting in no shrinkage in the wall when finally cast.

There are recommendations for minimum wall thickness of cast metals (S.C.R.A.T.A., in Beeley 1972:331); for instance in sand castings using magnesium alloy, the recommended wall thickness is 4.0 mm, and for sand castings using steel, the recommended wall thickness is 4.8 to 12.7 mm.

[6] The term 'riser' in the top of Figure 3.2 is synonymous with the term 'feeder'. Feeders are known as *risers* in America.

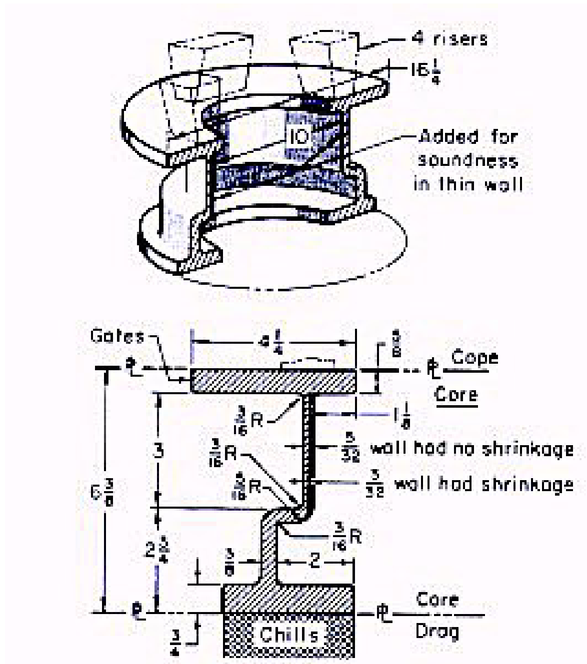


Figure 3.2 Increasing Wall Thickness to Eliminate Shrinkage Porosity (American Society of Metals 1962:28)

3.1.3. Design Problems Involving Junctions

Important areas for re-design are related to problems concerned with casting junctions (Casting Design Handbook 1962:49-56; Beeley 1972) such as the L, T and X-junctions, as shown in Figure 3.4. The main problem is that *hot spots* (essentially trapped molten metal) can form (Ravi and Srinivasan 1990; Campbell 1991). Hot spots arise when there is slower cooling at the intersection due to heat radiation from nearby elements. This is shown in Figure 3.3, where the hot spot shown as a red area and the arrows represent heat flows from the casting. The hot spot freezes in isolation in the casting process, leading to a cavity (a hole), in the casting junction. The Methods Engineer needs to take suitable remedial action by placing a chill or a feeder there, or by possibly modifying the shape of the joint by using fillets. Figure 3.4 shows the three fundamental intersection types (the L, T and X junctions) and how sharp angles in the junction can be re-designed to eliminate hot spots by fillet radii.

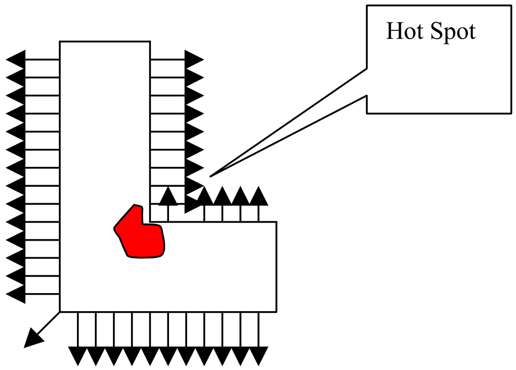


Figure 3.3. Effect of Junction Shape on Heat Radiation

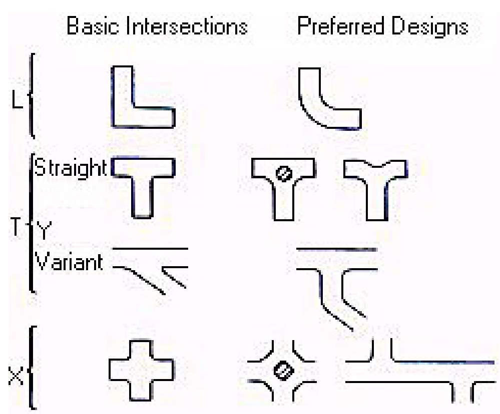


Figure 3.4 Redesign of Fundamental Junction Types to Eliminate Hot Spots (Beeley 1972:333)

3.2 Shape Decomposition

Casting geometry is of great importance as investigated in Section 3.1. A prime concern of Methods Engineering is thus being able to predict the solidification time irrespective of the shape of casting geometry. Chvorinov (1940, in Ravi and Srinivasan 1996) has established that the solidification time of any casting can be approximated by the ratio of volume and surface area that is involved in cooling, by the calculating its modulus. This is determined by the relation:

$$\boldsymbol{M = K(V/A)^2}$$

where V is the volume of the shape over the cooling surface area, A is the cooling surface area, and K is a constant depending on the thermal properties of cast metal and mould material.

The essence of the modulus process consists in componentising a casting into elementary geometric bodies and calculating the ratio volume/surface area for each basic body. To demonstrate, the modulus for a simple plate of length L, width W and thickness, T, is:

$$Volume = LWT\ mm^3$$

Cooling Surface Area = 2(LW + WT + LT) mm^2

Geometric Modulus, M,

= Volume/Cooling Surface Area

= LWT/2(LW + WT + LT) mm

For example, if 10kg of steel is cast as a sphere, with Volume = 1.3 dm^3 and Area = 4.3dm^2, the solidification time is approximately 11 minutes. If the steel was cast as a thin plate (with Volume = 1.3 dm and Area = 26 dm^2), the plate solidifies much faster (approximately 0.5 minutes) than 10kg of steel cast as a sphere. This is clearly because the heat contained in 10kg (that is, 13000 cm^3) is given off over a much larger surface area in the case of the plate, that is, the larger the heat emitting surface associated with a given volume, the faster the solidification.

A number of examples showing five elementary shapes for which there are empirical rules for calculating the modulus are shown in Table 3.1. Although not shown in the table, for each component there are rules to modify the modulus, due to such problems as radiation effects (handled by the *effective modulus*).

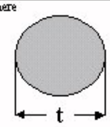
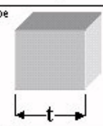
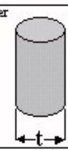
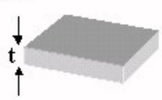
Shape	Volume V	Area A	Modulus V/A
Sphere 	$\Pi t^3/6$	Πt^2	t/6
Cube 	t^3	$6t^2$	t/6
Cylinder 	$\Pi t^3/4$	$3\Pi t^2/2$	t/6
Plate 	At	2A	t/2

Table 3.1 Moduli of Some Common Shapes (adapted from Beeley 1972:101)

Most castings are far more complicated than the ones shown in Table 3.1. Therefore, the system of substituted bodies is used, in which complex shapes are subdivided into simple basic components of equal modulus on the basis that two bodies of matching modulus solidifies in an equivalent time (Beeley 1972). Complex shapes are broken up into a set of standard components with well-defined moduli and each component has the cooling modulus applied to it allowing the freezing order of the shape to be ascertained. It should be borne in mind that this subdivision is only theoretical, and the interface between two basic components is certainly not a cooling surface, therefore it cannot enter into the calculation when determining surface area.

Using this decomponentisation concept, the Methods Engineer can qualitatively determine the likely solidification paths within the total shape. Solidification proceeds in the direction of increasing modulus to the feeder, which is the heat centre of a casting, and where all the shrinkage occurs (Sirilertworakul, Webster and Dean 1993b). After solidification, the feeder is removed from the casting as scrap metal.

The main advantage of the modulus is that it gives a rapid assessment that can be made at the engineering drawing stage as to whether a casting design might be castable (Bradley, Adams, Gadh and Mirle 1993; Cambell 1991), as well as giving a rough guide about feeder sizes (since a feeder must remain molten longer than the casting it is to feed).

In this research the componentisation process distinguishes between two sets of component type: those that define the structure (Ls, Ts, Xs) and those that join the first set together (bars and tapers). Numerous bespoke component types are also used, which are variations on the standard types. All of these components are of signficance in casting design (Casting Design Handbook 1962; Wlodower 1966; Campbell 1991).

Using the principle of shape decomposition, the Methods Engineer can assess the approximate solidification of a shape by breaking down the shape into a number of structural and joining components. This is demonstrated in Figure 3.5, which shows a cross-section componentised by sub-division into elementary components (adapted from Beeley 1972:101). A more detailed example of shape componentisation is shown in Appendix A Published Paper I, ‘Cast-Aid: A Decision Support System for Designing Castings’ (Mileman, Knight, Cowell and Preddy 1998).

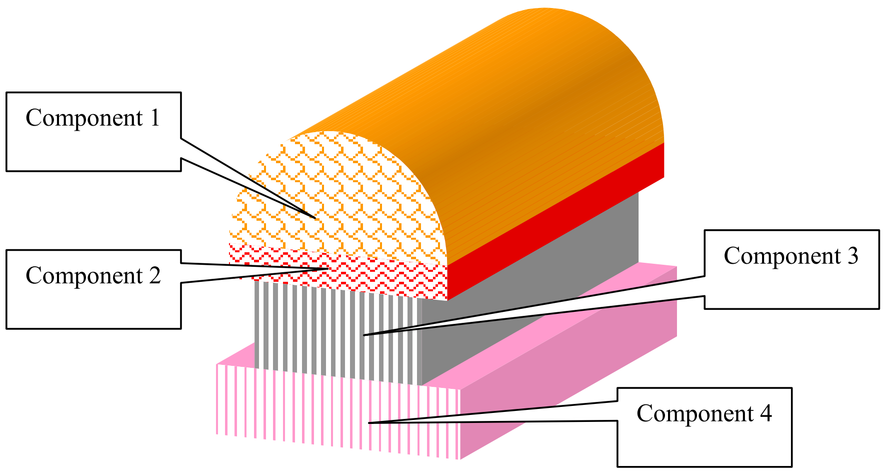


Figure 3.5 Cross-section Decomponentisation into Elementary Components (from Beeley 1972:101)

The modulus as a 'quick-see' solidification tool has a number of disadvantages, the fundamental one being is its oversimplification in numerical modelling terms; this is because the model does not take into account the complex nature of fluid flow. Furthermore, the model has little success in predicting dispersed micro-shrinkage, since it assumes that liquid metal is at a constant temperature in a mould. However, the modulus is a good engineering approximation and provides natural keys for shape retrieval. The model is particularly relevant since the Methods Engineer uses knowledge to chill, to feed and to modify problematic casting designs based on experience of previous shape componentisation.

Experience associated with how the components are joined is of crucial importance (Ravi and Srinivasan 1990). Consider for example the T-junction in Figure 3.6, which is made up of two plates. There is significant heat radiation from nearby elements and because of radiation at the joint shown, the cooling is slower. This leads to the formation of a hot spot, which as explained in Section 3.1.3 results in an isolated pocket of molten liquid that forms during freezing. Subsequent shrinkage of these cavities give rise to porosity and other casting defects. For this reason, arbitrary joins are not allowed, since these would lead to unsoundness. Methods Engineers therefore, would not decompose the T-junction in this way. Experience has confirmed that the central element shown in Figure 3.7, which includes fillets, is more sound (American Society of Metals 1962). Similarly, Y-junctions can often be re-designed as filleted T-junctions. A vast amount of experiential knowledge of casting design has developed around such decompositions and a library of components of good design helps to rule out the possibility of the design of an inherently uncastable shape (Knight, Cowell and Preddy 1995).

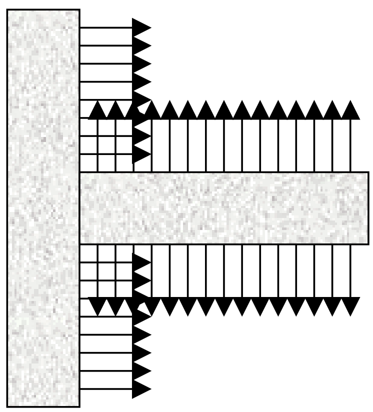


Figure 3.6 T-Junction Formed From Two Plates

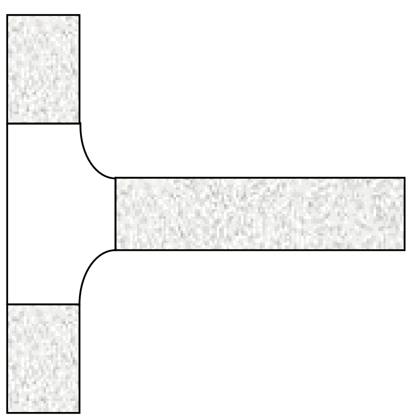


Figure 3.7 Filleted T-Junction

3.3 Existing Computer Based Approaches for Modelling Casting Shapes

The approach in this research has been to base the abstraction scheme to represent shapes using a decomposition of shapes that is specific to the casting industry. This representation draws on the expert's perception and reasoning process of casting. However, it is worthwhile to consider alternative approaches to modelling casting shapes and to judge whether these have been successful. There have been two approaches so far: Solid Modelling Techniques using CAD/CAM systems for solidification modelling and feature-based approaches usually coupled with knowledge-based systems for castability evaluation.

3.3.1 Solid Modelling Techniques

The traditional techniques used to model castings have included CAD/CAM systems generally coupled with solidification software to simulate the path of metal freezing. Solid modellers can (Jared and Dodsworth 1987:154):

> *"Hold a complete and unambiguous representation of the geometry of a wide range of objects."*

Solid models therefore provide a description of a shape that is *object focused* (Falcidieno and Giannini 1989). Techniques for shape representation in solid modelling fall into six general classes as shown in Table 3.2.

Pure primitive instancing
Generalised Sweeps
Spatial Occupancy Enumeration
Cellular Decomposition
Constructive Solid Geometry
Boundary Representation

Table 3.2 Methods used in Solid Modelling (Jared and Dodsworth 1987)

There have been a number of research projects in casting using some of these representation schemes; for example, in Constructive Solid Geometry (CSG) simple geometric primitives, such as block, cylinder, cone and sphere are used to build compound objects (Ballard and Brown 1982, in Hill 1990). These simple primitives are positioned and combined using the Boolean operators:

- Union ($\cup$)
- Difference (—)
- Intersection ($\cap$)

A compound shape, held in a binary tree (known as a CSG tree) is created hierarchically by combining objects using these basic set operations. Primitive objects are stored as leaf nodes, while Boolean operations are stored as interior nodes. If two solids A and B are combined with the Boolean operation op, the point set A op B comprises:

- For op = UNION: all points which are in A, B or both.
- For op = INTERSECTION: all points that are in both A and B.
- For op = DIFFERENCE: all points which are in A and not in B.

The main approach in current intelligent casting research has focused on feature extraction from solid modelling constructions, usually linked to a knowledge-based system for design assessment (You, Chu and Kashyap 1991). Although feature extraction has been accomplished based on a CSG tree structure, it has been problematic due to the non-uniqueness of CSG when it comes to extracting general feature information such as global shape information; this is imperative in castability evaluation (You *et al* 1991).

Luby, Dixon and Simmons (1988) tackle the problem by defining a shape grammar capable of design-creation by the use of a vocabulary of familiar geometric designs. The design is evaluated for manufacturability by the construction of the modulus model from the shape features.

Other approaches, such as Natarajan, Chu, Kashyap (1989) use a rudimentary geometrical solidification model to make a preliminary appraisement of castability for a restricted class of shapes. This is based on a cellular decomposition using the cooling modulus, rather than on whole features, somewhat like SOLSTAR (Corbett 1989).

3.3.2 Feature Modelling

Feature modelling is becoming recognised as a strong candidate for a single data representation for design, design analysis and manufacturing planning in the general context of total computer integrated manufacturing (Case and Gao 1993).

Bronsvoort and Jansen (1993), in their survey paper on feature modelling describe feature modelling as a relatively new approach in CAD/CAM, that:

> *"Allows the designer to model objects with elements that are on a higher-level, and closer to his way of thinking, than the lower level geometric elements used in solid modelling."*

Feature modelling makes information about a shape to be seen at a higher-level of abstraction, and not limited solely to its geometric model, as in solid modelling (hung, Patel, Cook and Simmons 1990; Bronsvoort, Bidarra, Dohmen, van Holland and Kraker 1996). Applications can use this functional higher-level information about a shape in several analysis and planning tools (van Holland and Bronsvoort 1997). The main work in casting has been to represent shapes using a feature-based approach, linked to a rule base for castability evaluation; as examined earlier in this chapter this approach has not been successful.

3.4 Section Slices

During the research it became apparent that Methods Engineers work from 2-D cross-sectional drawings of complex 3-D shapes; the rationale for this is that it is far easier for the engineer to understand a complex 3-D shape as a series of 2-D horizontal and vertical slices. Indeed, the literature of casting design concentrates heavily on using cross-sectional slices; see for example the Casting Design Handbook (1962). Methods Engineers study key slices from a feeding point of view using the process of shape decomposition, as examined in Section 3.2.

Although Kotschi and Plutshack (1981) have carried out research to simplify the evaluation of 3-D solidification by using a slicing technique used to simulate 3-D shapes as 2-D slices, their work however was focused on trying to lower the costs of simulation by using a less complex mathematical approach for casting solidification. The idea of using 2-D cross sections for shape retrieval is a different matter, an important finding that has been overlooked by previous research in applying CBR for the casting industry.

Representing a shape as a collection of 2-D rotationally symmetric slices (each of which is broken down using the casting modulus into elementary components) provides the key for developing retrieval mechanisms based on plane sections (see

Chapter 5 Similarity Metrics). To justify the use of cross-sections further, two engineering drawing examples are presented in the following sections.

3.4.1 Slice Example I

The first example is taken from the Casting Design Handbook (1962:46) of a permanent-mould cast aluminium piston. The engineering drawing of this is shown in Figure 3.8. The engineer has chosen two views through the drawing of horizontal and vertical cross-sectional slices: A-A and B-B respectively of the shape (shown in Figures 3.9 and 3.10). Given these slices the Methods Engineer would then decompose the slice into elementary components and calculate the modulus of each component. Using design knowledge of previous castings the Methods Engineer attempts to ensure that the new casting is sound. In this instance, the original design if cast would result in simultaneous freezing, leading to porosity and centreline shrinkage[7]. Once this problem was identified (essentially a wall thickness problem), the Methods Engineer took appropriate re-design measures. In this particular example, the casting was made sound by tapering the walls and adding ribs for metal feed paths to the bosses.

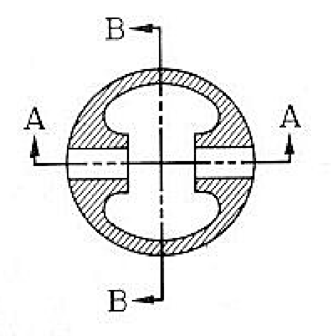
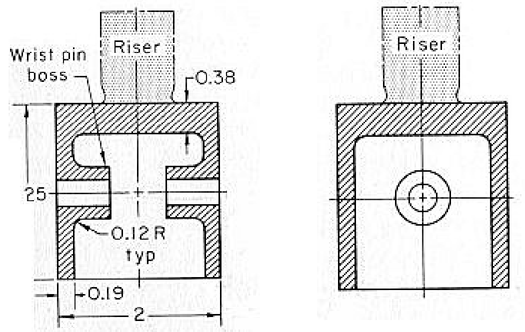


Figure 3.8 Design for an Aluminium Piston.

[7] *Centreline shrinkage* is a type of internal shrinkage cavity typically affecting the central zones of extended parallel walled sections; see Beeley (1972:209)



Figures 3.9 Horizontal (A-A) and Vertical (b-b) slices of piston from Figure 3.8

3.4.2 Slice Example II

The second example is of a rotationally symmetric sea-gland[8] shown in Figure 3.10 (Stone Foundries) to be cast in aluminium-bronze using a CO_2-Silicate mould.

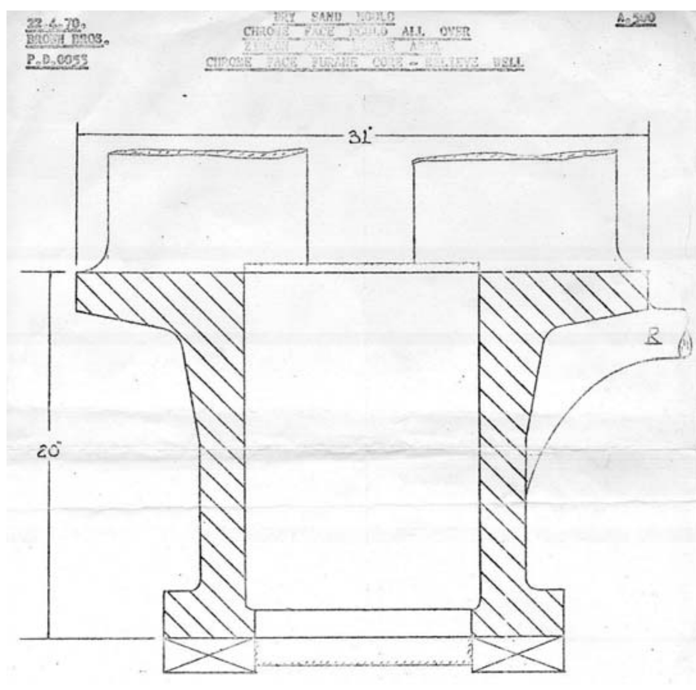


Figure 3.10 Sea-gland Engineering Drawing Section Slice

[8] A sea-gland is a casting used in the offshore industry for pumping sea water. The sea-gland shown in Figure 3.10 was cast by the Weir Pump foundry (*circa* 1970)

The engineer has componentised the shape using five components (Figure 3.11) each of which has a modulus value, shown in Table 3.3. A problem arises, because the top taper has a modulus of only 25 and freezes before the connecting L-junction (which has a modulus of 47). To solve this, the Methods Engineer, to ensure directional solidification, would place a central feeder and two chills at the lower foot to ensure porosity does not occur. This shape is the focus of the Published Paper I in Appendix A.

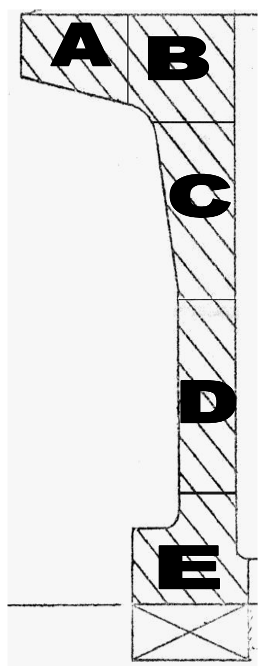


Figure 3.11 Sea-gland Componentisation into Elementary Shapes

Component	Type	Modulus
A	Taper	25
B	L-Junction	47
C	Taper	37
D	Bar	32
E	Flange	22

Table 3.3 Sea-gland Modulus Composition

3.5 Conclusions

The process of shape componentisation used by Methods Engineers has been identified. The componentisation technique is based on the system of substituted bodies using the modulus concept to determine the approximate solidification time of an arbitrary casting. It was seen that the componentisation process distinguishes between two sets of component types that are of signficance in casting design; those that define the structure, that is the junctions (such as L, T, and X) and those that join the first set together, that is the connectors (such as bar and taper). Much casting knowledge is associated with this shape decomponentisation.

Current computer-based approaches for modelling casting shapes were also examined in this chapter. The industry has mainly employed CAD solid modelling approaches for solidification analysis, because a complete solid model provides a description of a shape that is object focused. However, solid models do not give deeper knowledge about the shape at a higher-level of abstraction. For this primary reason feature-modelling is gaining interest as a way of modelling at a higher-level of abstraction, coupled with expert systems for castability analysis.

Methods Engineers work from 2-D cross-sectional drawings of complex 3-D shapes and some examples were given to illustrate this in real casting design. Much experiential knowledge of castability has grown up around such decompositions and knowledge of re-design is often linked to this component classification. Shape retrieval based on plane sections, corresponding to the way Methods Engineers reason about solid objects is therefore what should be tested in the evaluation of the approach (see Chapter 6). Before this, the following chapter examines the abstraction of the data-model based on the shape componentisation process.

Chapter 4 Data Model Based on Componentisation

This chapter investigates how the data model for a shape componentisation is produced. The previous chapter determined that the Methods Engineer expertise is based on a unique decomposition of the shape to be cast into well-known components with well-defined moduli, and that complex 3-D shapes are represented as 2-D cross sections. A novel contribution of the research has been in identifying and abstracting this shape decomposition technique. This serves as a way of representing shapes from which the abstraction of similarity metrics can be achieved.

There are many advantages in constructing a modulus model which emulates the Methods Engineering process, particularly when considering that much experiential knowledge of castability has grown up around such decompositions. Knowledge of re-design is frequently linked to this component classification. Previous shape representation approaches have used an arbitrary decomposition followed by a numerical estimation of the element moduli. As Chapter 3 investigated, this shape componentisation scheme corresponds to the way Methods Engineers reason about solid objects. Abstracting this process lead to a series of similarity metrics; this is discussed in the following chapter.

Overview of chapter

Section 4.1 investigates the basic primitives used in the shape decomponentisation process. Section 4.2 looks at how complex 3-D shapes are represented using these elementary components. In Section 4.3, it is shown that casting shapes can be represented as graphs. In Section 4.4, the need for orientational information to counter ambiguities is examined. In Section 4.5, the data model for shape representation used in this research is determined, and examples are given of the data model.

4.1 Basic Primitives for Shape Componentisation

Castings are represented using six component types identified from knowledge elicitation, with Chapter 3 setting out how these basic primitives are used in the shape componentisation scheme. In this research, components have been split into two groups: **Structure** defining components (the junctions) and **Connector** components (such as bars and tapers) that join the first set together.

The six components are a rich enough set that can enable the representation of a range of complex castings. That this set is sufficient is justified in the evaluation section of this thesis, Chapter 6 Evaluation, in that a case base of one hundred castings could be represented using these six components, and furthermore, as the test results show, this set of components is sufficiently powerful for representing a range of castings from the domain of castings with rotational symmetry.

A further rationalisation to strengthen this claim is found in the casting literature (see for example: Beeley 1972; MOD 1979; Campbell 1991); the same basic set of components appears. Having identified this set, a contribution was made in distinguishing two types: a range of junctions (cross, L, T), the structural components, and connecting components (bar, taper).

There is no need for a massive set of curved components, because these components can be treated as bars (Beeley 1972); this is explained in more detail below. Previous research using design-by feature has not focused on representing castings as 2-D cross sections. Instead, shapes are represented as a collection of 3-D components. The flaw in this approach is that many thousands of components are required to represent a good test set of castings. Previous researchers have found that when using 3-D components there is a danger of a combinatorial explosion in the number of primitives in the component library (Luby, Dixen, Simmons 1988).

The structure defining components are shown in Figure 4.1 (clockwise: T-Junction, X-Junction, Flange, L-Junction,), and the connector components are shown in Figure 4.2 (bar, taper).

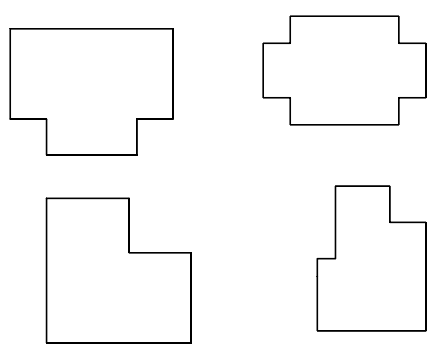


Figure 4.1 Structure Defining Components

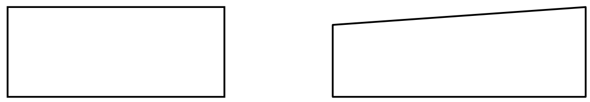


Figure 4.2 Connector Components (Bar and Taper)

4.1.1 Curved Components

These six basic components are extensible; for example, the semi-circular taper in Figure 3.5 of Chapter 3 is not included in the basic set, but it could be included as another component. It might be supposed that an extensive assortment of angular or curved components is necessary (since most casting are intricate). However, two bodies of equivalent modulus solidifies in the same time, thus angular and curved components of the same ruling dimension have the same modulus; for instance (Beeley 1972), the sphere, cylinder and cube all have moduli of:

$$t/6$$

where **t** = thickness. Similarly, rings and hollow cylinders can be treated as bars or plates, which they would form if opened out (Beeley 1972:100). For the T-junction connected to a curved bar in Figure 4.3, which represents a 3-D object, this can therefore be treated as a T-junction joined to a bar, as shown in Figure 4.4.

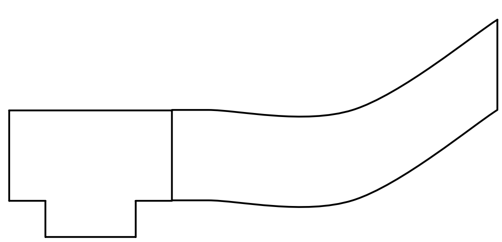


Figure 4.3 T-junction Connected to Curved Bar

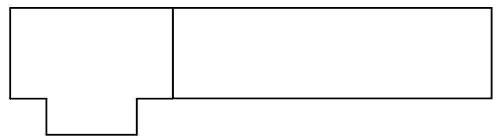


Figure 4.4 Substituted Body of Equivalent Modulus of Figure 4.3.

4.1.2 Bespoke Components

During knowledge elicitation it was observed that the Methods Engineer referred to *bespoke components*. These are considered as variations on the basic six components, allowing a richer classification of shapes. In this research, the Bespoke-Taper and Bespoke-T-Junction have been incorporated, as shown in Figure 4.5. Bespoke components *are* components, but are distinguished so that the Methods Engineer knows that a particular component is modified slightly; for example, consider the bespoke T-junction in Figure 4.5. It is seen that the right-hand side of the base is sloping.

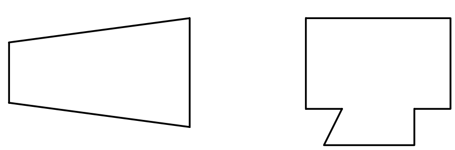


Figure 4.5 Bespoke Components (Taper and T-junction)

4.2 The Treatment of Complex 3-D Shapes

Castings are complex 3-D shapes, yet so far in this thesis castings have only been considered as a collection of 2-D sections. Initially, this might be considered as a deficiency in the representation scheme. However, during the process of knowledge acquisition and in particular in interviewing Methods Engineers, it was learnt that experts referred to crucial 2-D cross-sectional views that would trigger off design knowledge. Therefore, the problem of shape representation can be reviewed as one in which shape componentisation is characterised as an assemblage of elementary 2-D section slices, coupled with key design knowledge about each section slice. This corresponds closely to the way Methods Engineers reason about solid objects and agrees with the literature of casting design (Casting Design Handbook 1962).

To show how shapes are treated as 2-D objects in the representation scheme, it is first necessary to distinguish between axi-symmetrical and arbitrary 3-D shapes. In the research, it was established that there are two groups of objects:

- **Object Group 1:** Extruded shapes and axi-symmetrical shapes

- **Object Group 2:** Arbitrary 3-D shapes

4.2.1 Object Group 1

Extruded objects, such as the one shown in Figure 4.6 can be treated as two bars and an L-junction (Figure 4.7). The reason why Figure 4.6 is not regarded simply as an L-shaped component is that Methods Engineers would not decompose the L shape that way since the L shape is neither physically achievable nor indeed desirable. This is because there is slower cooling at the joint due to heat radiation and thus the joint causes a *hot spot* (see Chapter 3 - Section 3.1.3 for more information about hot spots), which freezes in isolation in the casting process. Methods Engineers embrace the breakdown as shown in Figure 4.7 as a filleted L-junction and two bars. The essential difference here is that there are no significant heat radiation effects to be taken into account.

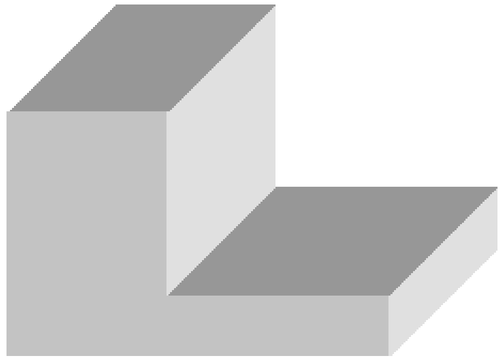


Figure 4.6 Extruded plates

Figure 4.7 The Corresponding 2-D Structure of Figure 4.6 as Bar, L, Bar

A second example is the engineering drawing for a flywheel (Figure 4.8). Its componentised horizontal 2-D cross section is shown in Figure 4.9.

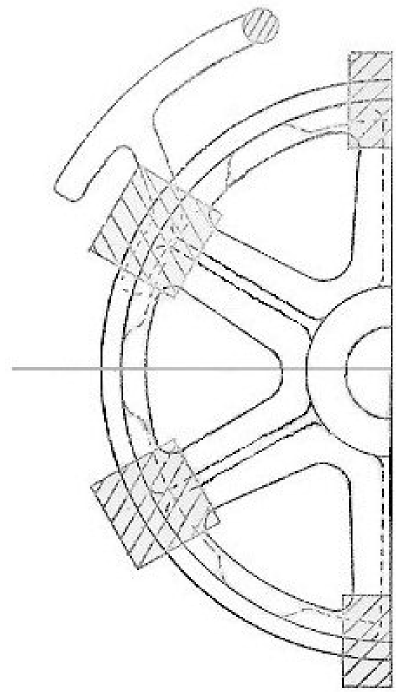


Figure 4.8 Engineering Drawing of a Wheel (Stone Foundries)

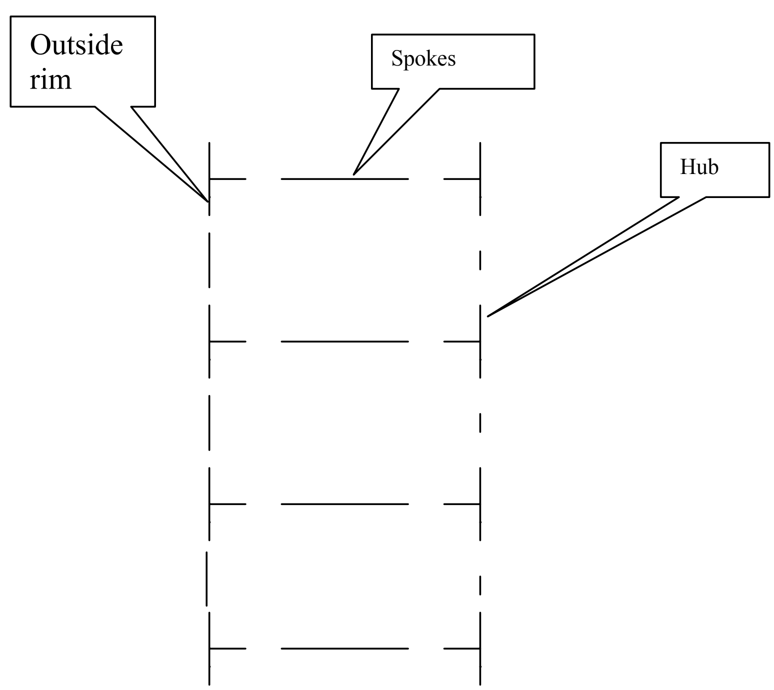


Figure 4.9 Graphical Representation of the Wheel (in Figure 4.8) as T-junctions and Bars

4.2.2 Object Group 2

Arbitrary 3-D shapes, taking as an example the mug in Figure 4.10, can be treated as the two cross-sections as depicted in Figures 4.11 and 4.12. These two 'views' of the mug can provide valuable identifiers to enable accurate retrieval.

Although arbitrary 3-D shapes are not the focus of this research, in Chapter 7 - Conclusions, arbitrary 3-D shapes are included in an agenda for future work. The representation used is only an abstraction, but this abstraction must tie in to the crux of this thesis, that is to retrieve similar shapes from a casting point of view. The representation as a set of 2-D sections does give such an abstraction and one that is to be tested for its efficacy for retrieving abstractions.

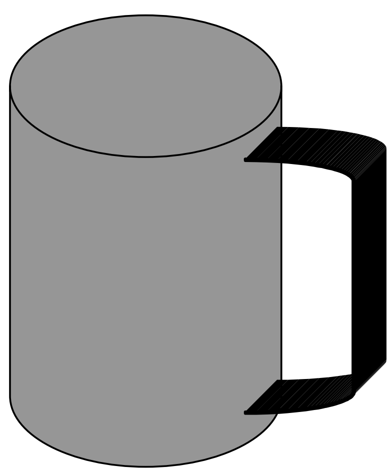


Figure 4.10 Arbitrary 3-D Shape (mug)

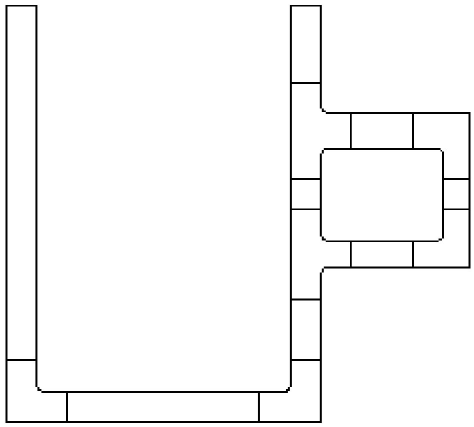


Figure 4.11 2-D View 1 of Mug (of Figure 4.10) made of Bars, L-junctions and T-Junctions

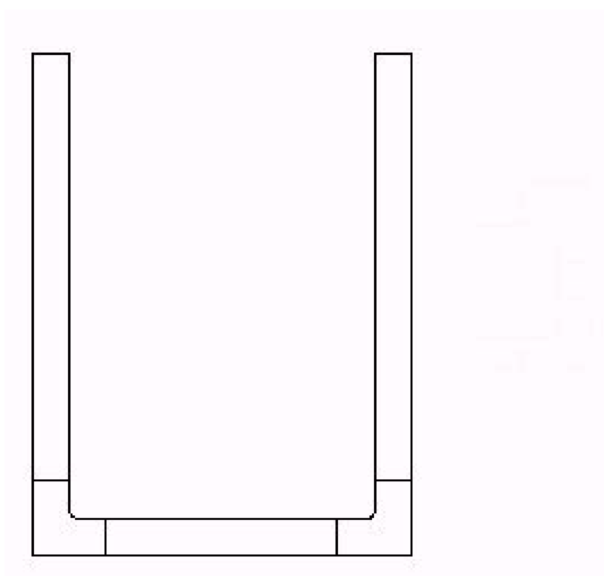


Figure 4.12 2-D View 2 of Mug.

4.3 Graph Structure for Shape Representation

The representation of casting shapes used in the research is a novel one, and takes the form of a graph structure, where nodes are the components and the arcs connect components. Shapes are essentially an ordered list of objects attached at specified connection points; for example, the two ends of a bar, the three ends of a T. These lists completely specify the element network. An example shape is shown in Figure 4.13, and its graph is shown in Figure 4.14. Each node in the graph corresponds to a component, and the arcs act as interfaces (another object) between components; this is explained further in Section 4.5.

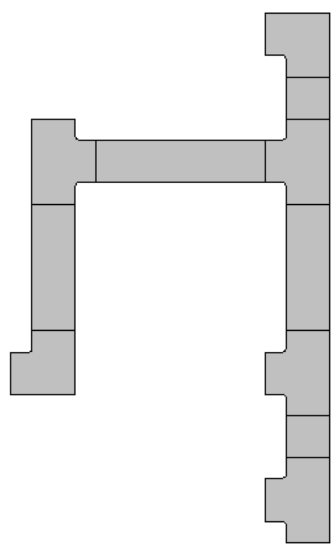


Figure 4.13 A Casting Made of 3 Component Types (Bar, L, T)

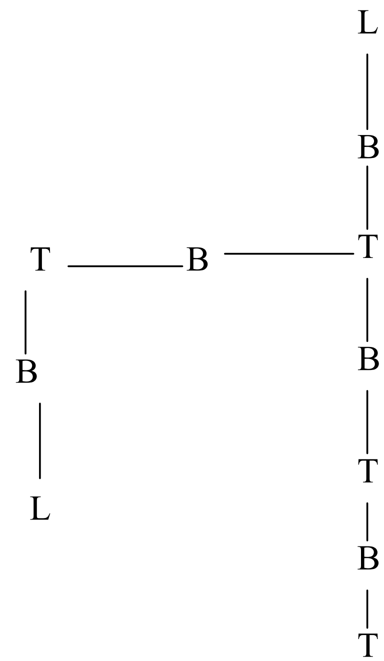


Figure 4.14 Graph of Casting Slice of Figure 4.13

4.4 Orientation

The graph in Figure 4.14 is an abstraction of the shape depicted in Figure 4.13, which can be used as a basis for matching with other shapes. However, the graphical representation is insufficient in that it does not contain complete information as to orientation of the various joining components to one another or to the axis of rotation. For example, if the shape represented in Figure 4.13 is rotated by an arbitrary degree around the x or y-axis, it still has the same graph structure as Figure 4.14, although the overall 3-D shape is very different. The problem is that, for example, although an L is joined to another L via a bar, it is not known whether it is joined in the left-hand or the right-hand sense. This leads to some ambiguity in the orientations of the represented structures. Hence, although metrics based on the basic graph allows shape retrieval of similar shapes with similar orientations, it also allows retrieval of shapes with differently orientated structures.

To tackle this problem, an improvement on the graph structure was examined during the course of this research and a solution offered by the introduction of orientational information:

1. The first improvement may be made if the convention is adopted that for the 2-D section (including the axis of rotation), the axis is shown to the right and pointing "north". For sections perpendicular to the axis of rotation, an arbitrary "north" can be chosen, since in this case it is only the relative orientations of components which is of importance.

2. Adding a bearing to each of the junction sections can make a further improvement. Figure 4.15 shows an L and T section at bearings 0^o, 90^o, 180^o, 270^o, and at an arbitrary bearing of Φ^o.

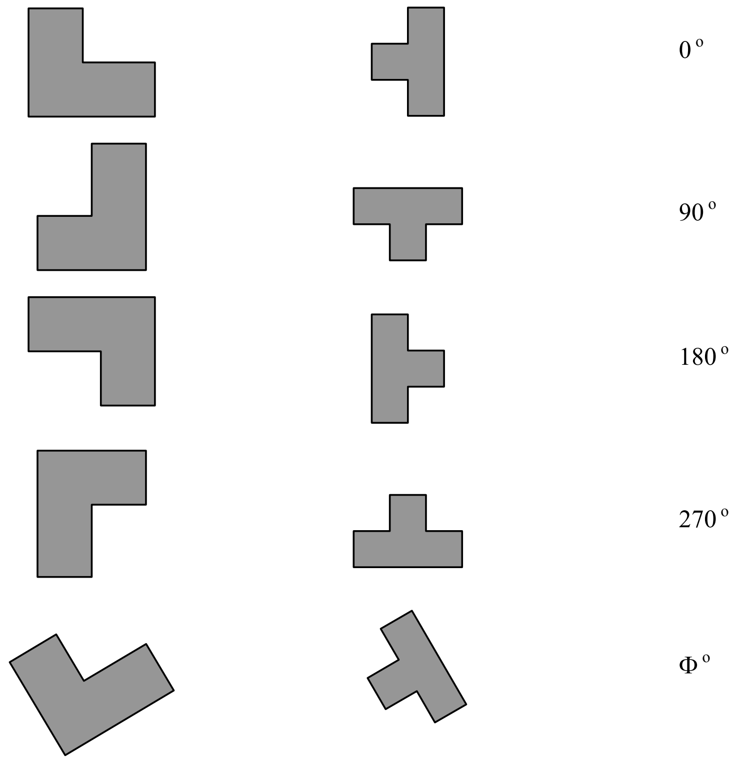


Figure 4.15 L and T section at Various and Arbitrary Bearings

4.5 Data Model for Casting Shapes

A comprehensive data model for representing casting shapes in the decomposition scheme as set out in Chapter 3 is shown in Figure 4.16. A 3-D casting *S* is represented as an object having a number of 2-D cross-sections, where each slice can either be horizontal and vertical. As stated in Chapter 1, in this research shapes are limited to just one section slice, but the abstract model allows *N* section slices. Each of these 2-D slices is made up of parameterised 2-D components, with interfaces on each component allowing connectivity to other components. Each component corresponds to a node in a graph, and interfaces are arcs that connect nodes.

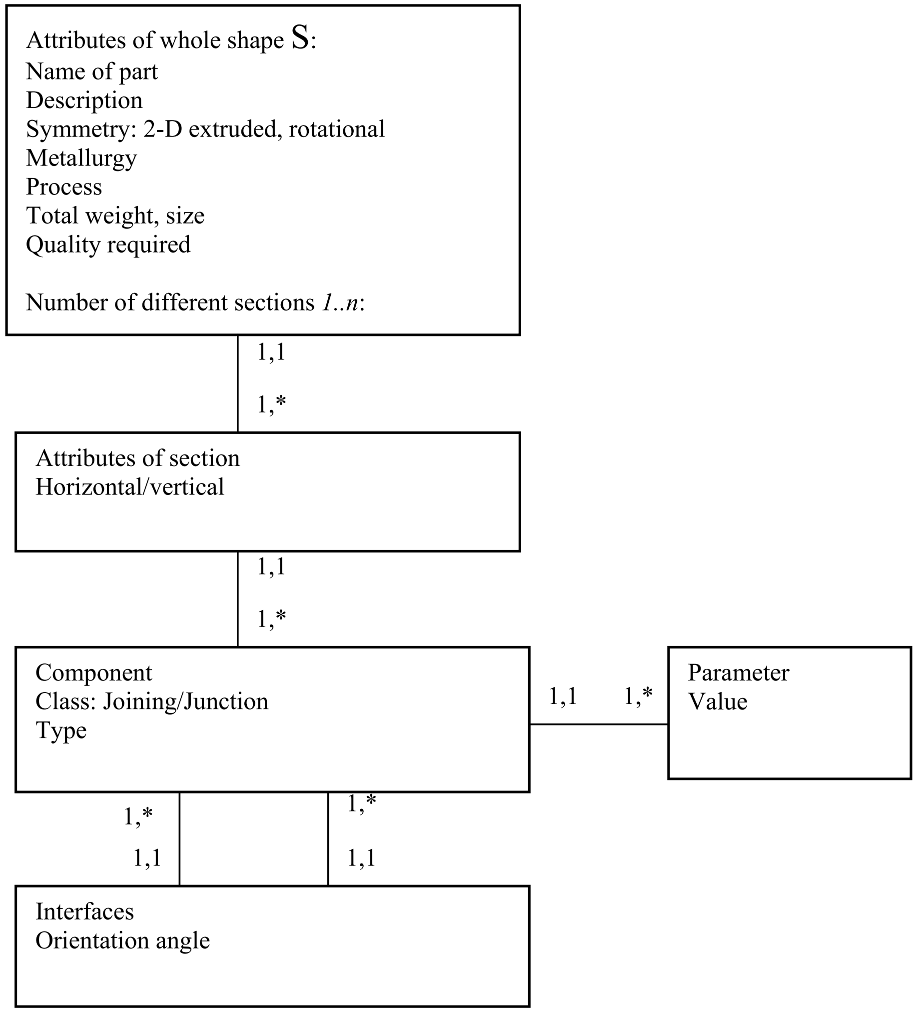


Figure 4.16 Data Model for Casting Shapes[9]

[9] A one-to-many relationship is shown by 1,1:1,*

The basic components have a uniform representation, namely

Component (Identifier, Type, Geometry, Location, Number of Interfaces, Chills, Feeders)

where *Identifier* is a symbol uniquely identifying a given component. *Type* is one of: Bar, Taper, L-Junction, T-Junction, X-Junction, Flange, Bespoke-Taper, Bespoke-T-Junction. *Geometry* is a list of parameters specifying the component geometry: for a bar [*Thickness, Length*]; for an L or a T-Junction [*Base Thickness, Arm Thickness, Fillet Radius*]. *Location* is a point in 3-D space, using the vector *O,X,Y,Z* (Plastock and Kalley 1986). *Chills* and *Feeders* contain information about where feeders and chills are located on the component. *Modulus* is the modulus calculation for each component type, taking into account the geometry of the component and its neighbouring components.

Similarly, interfaces have a uniform representation, namely

Interface (Identifier, Interface Number, Location, Connects Identifier, Connects Interface Number, Angle)

where *Identifier* refers to the component Identifier the interface is owned by. *Interface Number* is an ordinal value of the interface. *Location* is a point in 3-D space of the interface, using the vector *O,X,Y,Z* (Plastock and Kalley 1986). *Connects Identifier* refers to a component Identifier the interface is connected to. *Connects Interface Number* refers to an interface Number on the connecting component that the interface is connected to. *Angle* is the orientation of the interface: 0 or 180°.

An implementation of the data model was achieved using *AutoLisp*[10] (AutoLisp, 1994), in a prototype system developed at Greenwich University called *CastAID* (Preddy, Knight, Cowell and Mileman 1997) and this system can allow shapes to be drawn using the shape componentisation scheme, and saved in terms of component and interface objects for that shape. The *AutoLisp* code for the component and

---

[10] A programming language based on Lisp that comes packaged with AutoCAD Release 13 (Autodesk Inc, 1994). The AutoLISP Reference can be found in Customisation Guide, pp: 159-578

interface definitions are given in Appendix F. Examples are given in Appendix F of this data model, and an actual computer representation of a complete casting of the Sea Gland of Chapter 3 - 3.12.

4.6 Conclusions

The abstraction scheme used to represent shapes was investigated, and this scheme gives a way for the abstraction of similarity metrics. During knowledge elicitation it was ascertained that a decomposition of shapes specific to the casting industry already existed in practice and a unique representation could be derived from this traditional decomposition. The decomposition, based on set of component types of significance in casting design is a fundamental one used over many years by Methods Engineers. The main advantage over other proposed techniques is that the decomposition is related more closely to the way Methods Engineers reason about shapes, particularly in that design knowledge of casting is linked to this decomposition and component features. This abstract structure is graphical in nature, providing keys for pattern recognition techniques to apply to it.

There are a number of different problems associated with casting a shape, and in Section 3.1.1 these were taken to be decisions on the correct orientation, advice on castability and the number and the position of feeders and chills, each of which is connected with a different structural feature.

The next chapter examines and outlines a series of metrics that can be applied to the graphical structure of a casting; the goal of which is to use these metrics in prototype CBR system for shape retrieval.

Chapter 5 Similarity Metrics

This chapter investigates the retrieval of casting shapes by developing search algorithms for close partial matches. Unlike standard database retrieval where queries either match a record exactly or not at all, the problem faced in this research is that no distinct casting shape is likely to be identical to another. Kolodner (1993) gives an example that can be used to highlight this problem: consider a database with three records that have only the single field 'colour', and each respective record is instantiated with the values red, blue and yellow. If the query submitted is 'red', only the record 'red' is retrieved. If the query submitted is 'orange' or 'fuchsia' then no records are retrieved, although both colours are close to red. Shape matching therefore, should be directed towards records that have a high degree of relevance to the target shape that is to be matched. To achieve this, more sophisticated algorithms than the ones used in standard database retrieval need to be developed (Raphael and Kumar 1995).

The similarity measures that have been developed in the research are based on features extracted from the structural graphs that represent castings. Structural graphs are made up of a set of nodes and a set of edges connecting two nodes each (Gebhardt, VoB, Gräther and Schmidt-Belz 1997), where every node embodies structural information about a shape; for example, component type, number of free interfaces, whether the component is a leaf node and so on. Perfect similarity between casting shapes S1 and S2 is obtained when they have identical data models. However, for graphs that do not match completely there are a number of features that can be extracted and compared; these features are used to form search indices. These chosen features are calculated individually, and finally combined into a single similarity metric that gives a percentage match of closeness of a target case to a retrieved case. Some of the features that can be matched on are on the types of component (such as bar, junction) and shape assemblage, using the maximum common subgraph.

Overview of chapter

Section 5.1 investigates similarity metrics and details four shape similarity metrics for feature extraction from similarity graphs, giving reasons why these were chosen in the research, and their shortcomings. Metrics were devised for: components, MCS (Maximum Common Subgraph), leaves and cycles. In Section 5.2 a generalised similarity metric is given, and this represents a weighted-sum of the similarity measures based on the different features extracted from two graphs.

5.1 Proposed Similarity Metrics

Similarity metrics perform a central role in CBR (Brown and Filer 1995; Wang, Ishii 1997). This is because in CBR a problem is solved by distinguishing its similarity to a known problem (that is, a *case*) and then adapting the solution to solve the new problem. Ferguson and Bridge (1999) describe similarity metrics as:

"An operator that when applied to two objects of type $\propto$, returns a number, usually a real from [0,1], denoting their degree of similarity."

In the CBR literature, similarity metrics have typically been of two types: those that return a Boolean-value or those that return a real value in which there is a degree of similarity (Bridge 1998); the latter type is more natural and corresponds to the concept of *degrees* of similarity. Bridge goes on to examine several ways of computing the similarity of object representations, including feature-based, geometric and structural approaches. The last approach is the one taken in this research, in which similarity is based on graph matching where nodes denote objects and edges are relations between objects.

A similarity measure σ(S1, S2) is proposed in Section 5.2, which takes a weighted set of features extracted from graphs S1 and S2 that represent shapes. The result of this measure is a score between zero and one indicating how close S1 is to S2, based on the *nearest neighbour* matching function (Kolodner 1993). The higher this score is, the greater the resemblance between objects S1 and S2. For a perfect match, the value one is returned, while zero is a perfect mismatch. This function takes the basic form of (Watson 1996):

$$Sim(T,S) = \left[\sum_{i=1}^{n} f(T_i, S_i) * w_i \right]$$

where *T* and *S* are objects; *n* is the number of attributes in each object; *i* is an individual attribute from 1 to *n*; *f* is a similarity function of attributes in objects *T* and *S*; and *w* is the importance weighting of attribute *i*.

It is necessary to determine which features of an object can be extracted and used in the similarity measure. This problem is known as the *indexing problem* (Kolodner 1993; Taylor 1997). Essentially, the task is to determine the distinguishing attributes of an object; that is, which factors differentiate one object from another.

A set of features was identified in this research, and these are:

I. The type of components that make up a shape.
II. The Maximum Common Subgraph (MCS).
III. The leaf nodes of a shape: that is, those components joined to not more than one other component.
IV. The number of cycles in a shape.

Similarity metrics have been devised for each of these extracted features, as discussed in the following sections 5.1.1 to 5.1.4. Particular emphasis is placed on the validity of choosing each metric and the likely efficacy the metric has towards the goal of shape retrieval. It is the goal of Chapter 6 to evaluate the efficacy of these metrics with respect to different casting problem investigated. As stated in Chapter 1, the casting problems that solutions are desired for are:

- Correct orientation (moulding direction) of a shape
- Number of position of feeders and chills
- Design advice (including special problems encountered with a shape)

5.1.1 Matching Component Types

The approach presented in this research decomposes castings using various components of two types (connectors and junctions; see Section 3.5). The proposed first metric is based on the assumption that two shapes are similar if both shapes *share* similar components.

The "component" metric operates by taking a target shape S1, and a retrieved shape S2, and determining how many components in S1 *share* matching components in S2. For instance, consider shape S1, with components BB, and shape S2 with components BBB. Taking shape S1, the number of shared 'B' components is two; that is, two 'B' components are shared between shapes S1 and S2. Shape S1 therefore shares 100% of its components with S2. Comparing S2 to S1, only 2/3 (75%) of shape S2 is shared with S1; thus to gain the correct shared component value the metric results are multiplied, that is: (2/2)*(2/3); otherwise a 100% match would be returned if S1 was only compared to S2.

As an example to make this clearer, consider shape 1a and shape 2a in Figure 5.1. The component metric compares shape 1a to shape 2a, and the results of this are shown in Tables 5.1 and 5.2, showing the component type, number of components and the shared component match of shape 1a to shape 2a.

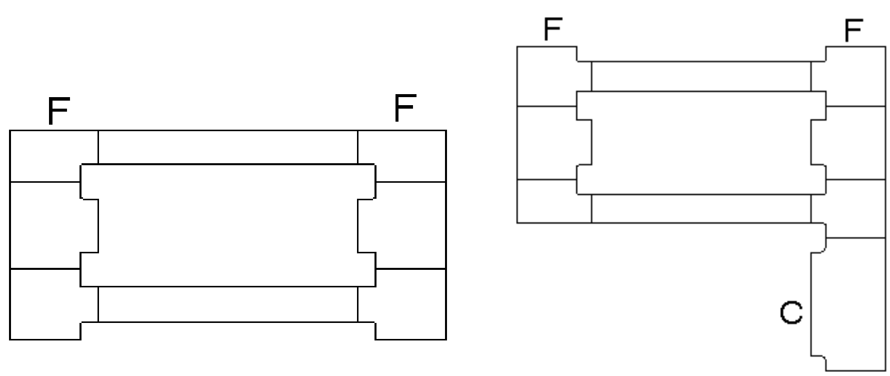


Figure 5.1 Shape 1a and Shape 2a (left to right respectively)

Shape 1a		
Component Type	Number of Component Type	How many components in Shape 1a match Shape 2a
L	2	1/8
T	4	4/8
B	2	2/8
TOTALS	8	7/8

TABLE 5.1 Component Analysis of Shape 1a

Shape 2a		
Component Type	Number of Component Type	How many components in Shape 2a match Shape 1a
L	1	1/9
T	6	4/9
B	2	2/9
TOTALS	9	7/9

TABLE 5.2 Component Analysis of Shape 2a

The final component match between the shapes 1a and 2a is 68%; that is, shape 1a shares 7/8 of its components with those of shape 2a. Similarly, shape 2a shares 7/9 of its components with shape 1a. The product of these shared component percentages gives an overall component match of (7/8)*(7/9) = 68%.

Pseudo-code for the component metric is shown in Table 5.3, and was implemented in *AutoLisp* (see Appendix E for the code).

```
; CBR component-metric
; compares shape1 (s1) with shape2 (s2) returning percentage
; of similarity of components;

Component-Metric % = is-in(s1,s2) * is-in(s2,s1)

; given by the function:


FUNCTION is-in (s1,s2) RETURN shared-components

    shared-components = 0
    ; for all components in shape s1, calculate how many of type c, are in
shape2

    FOR EACH component c in s
          shared-components = shared-components + CALL fit (s1,c, s2)
    END FOR

   RETURN shared-components / LENGTH (s1)

END FUNCTION
```

TABLE 5.3 Component Metric Pseudo-Code

Although a target shape may produce a close match in componentisation, other factors may also be important, such as the spatial layout of the shape, which includes meaningful information about component connectivity. This is not taken into account in the component metric. Another area for improving the efficacy of the metric is one of abstraction; for example, two connected bars can be treated as one bar, and similarly, a long thin taper can be treated as a long thin bar. To be able to abstract shapes would be a valuable addition to making the metric more sophisticated. This topic is returned to in Chapter 7 – Conclusions.

5.1.2 Maximum Common Subgraph (MCS)

A fundamental similarity measure when cases are represented as graphs is the size of the largest, or the maximum matching subgraph (Gebhardt 1997). This is not surprising when considering that valuable information can be gleaned from how individual components are connected spatially. For example, Figure 5.2 shows two shapes that have identical components, scoring 100% in terms of their component similarity. However, there are subtle differences when the shape connectivity is considered.

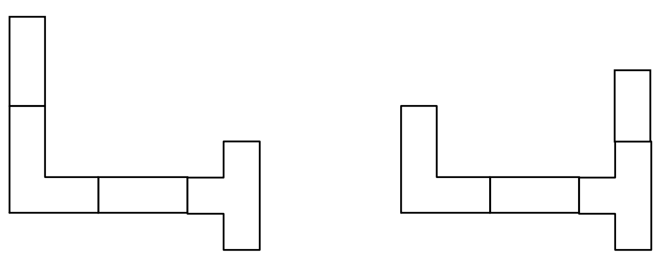


Figure 5.2 Shapes with Same Component Count, but Different Spatial Layouts

Therefore, a natural similarity measure is the size of the maximum common subgraph (MCS), which is defined by the largest graph contained in a set of graphs. Gebhardt *et al* (1997) provide an example showing that the MCS of three graphs a, b, c is graph d Figure 5.3); graph d is contained in the original three graphs (and it is the *largest* such graph).

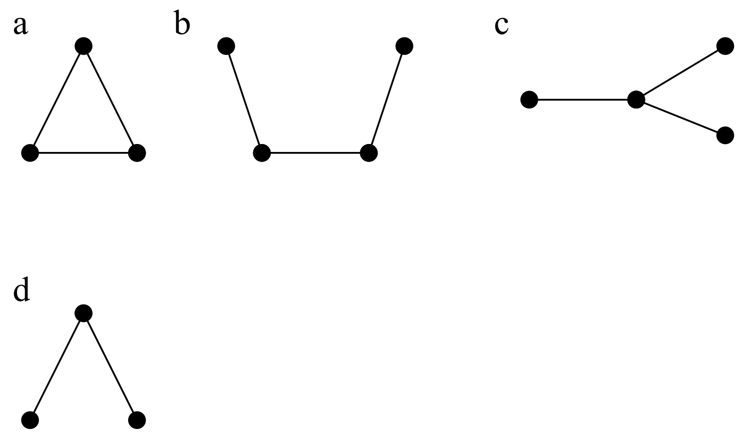


Figure 5.3 MCS of graphs a, b, c is graph d (Gebhardt *et al* 1997)

The MCS metric was implemented in *Visual Basic* (See Appendix D for the code). The algorithm starts with two graphs (A and B) as its input. It then looks for every combination of nodes X and Y; X from graph A and Y from graph B. It initially checks to see if the selected pair of nodes match. If they match then the algorithm calculates all possible combinations of neighbouring nodes (X_n, Y_n) not already visited by recursively searching these subnodes within this branch of the recursive search and tries to match them using recursion. When the two nodes X_n and Y_n do not match, the maximum remaining common subgraph from this pair of nodes is set at zero and the algorithm returns the remaining subgraph from the start pair of nodes (X, Y). The algorithm returns the maximum common subgraph by reporting the biggest common subgraph that it encountered while searching for each initial choice of matching nodes. Pseudo-code for the MCS Metric is shown in Table 5.4.

```
FUNCTION match(Graph A, Graph B, Node x, Node y) RETURNS MCS

IF x type matches y type THEN
        Calculate all possible combinations of neighbouring nodes not already
visited
        Mark current nodes as visited on the graphs for this path
        FOR EACH combination of neighbouring nodes (x1,y1)
                CALL match(Graph A, Graph B, node x1, node y1) ; Recursion
        END FOR
        Calculate remaining MCS
ELSE
        MCS = 0
END IF

Return MCS

END FUNCTION
```

Table 5.4 Maximum Common Subgraph Pseudo-Code

5.1.3 Leaves

A comparison of shape leaf nodes may be a useful metric and one that is evaluated in Chapter 6. The leaf metric is defined by the nodes of a graph which are connecting components (for example: bar, taper) and are those components joined to not more than one other component. The metric effectively gives a measure of the 'spikiness' of a shape. The unshaded bars in Figure 5.4 have been highlighted to show the leaf nodes, with shape *a* having 3 leaf nodes, and shape *b* having 4 leaf nodes. The match between shape *a* and shape *b*, is 75%; that is, the number of leaves in shape *a* divided by the maximum leaves in shape *a* and shape *b*.

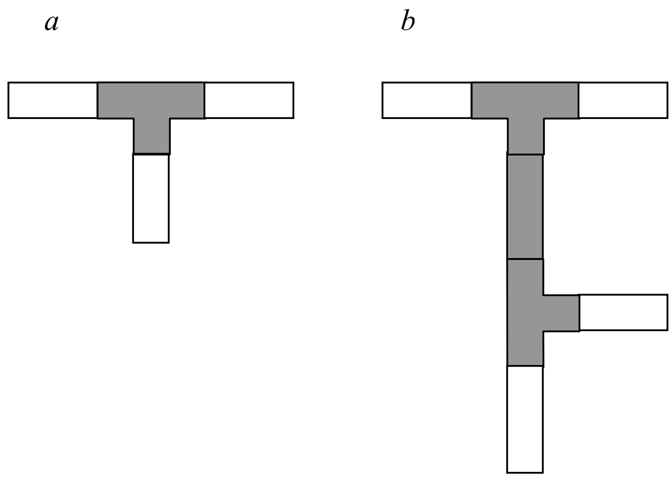


Figure 5.4 The Leaf Nodes of Shapes *a* and *b* are the Unshaded Bars

5.1.4 Cycles

Cycles are defined as *loops* within a shape. Consider Figure 5.1. There is one cycle in each of the shapes 1a and 2a, since the components in both shapes form one fully enclosed loop. These shapes match by 100% solely on having identical cycles. The cycle metric uses a simple count of the number of cycles in a shape. This characteristic of a shape may provide a useful measure of shape difference. However, because shapes may have, for example, three matching cycles each, does not infer the shapes are a good match. As the component metric showed, a shape may have similar components but may have a very different spatial layout. Similarly, two shapes may have the same number of cycles, but may have different spatial layouts and thus may not be a good match. Chapter 7 evaluates the cycle metric to determine its efficacy in shape retrieval.

5.2 Generalising the Similarity Metrics

All of the proposed metrics in Section 5.1 can be generalised as a similarity measure $\sigma(S1,S2)$ between shapes S1 and S2. This measure represents a weighted sum of the similarity measures based on different features extracted from the graphs of S1 and S2:

$$\sigma(S1,S2) = w_{comp}\sigma_{comp} + w_{mcs}\sigma_{mcs} + w_{cycle}\sigma_{cycle} + w_{leaf}\sigma_{leaf}$$

Variation of the weights in this formula allows a general test of retrieval against any given casting problem. How the optimal weights were determined is examined in Chapter 6 – Section 6.4.2.

The individual similarity metrics in Section 5.1 are defined in the following sections.

5.2.1 Component Metric

- $\sigma_{comp}(S1,S2)$ is a measure based on the number of component types that are common to the two graphs S1 and S2. If S1 and S2 are nearly identical, σ_{comp} is close to 1. If S1 and S2 are not identical σ_{comp} =0. This metric is given by:

$$\sigma_{comp}(S1,S2) = \left[\sum MIN(N(comp,S1),N(comp,S2))\right]^2 \Big/ N(comp,S1)N(comp,S2)$$

where N(comptype, S) = number of components of type comptype in S

For example, shapes S1 and S2 in Figure 5.5 do not match and the metric returns a zero score. For a partial match, such as shapes S1 and S2 in Figure 5.6, the metric returns 25%; that is, (2/8)*(2/2) = 25%, since there is not a perfect component match.

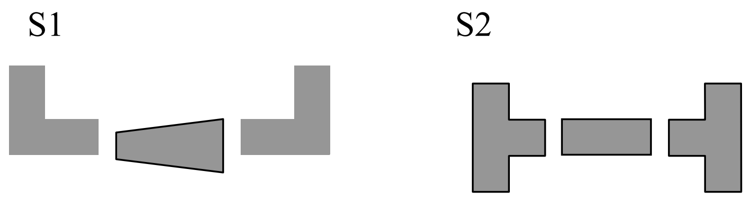


Figure 5.5 No Component Correspondence Between S1 and S2

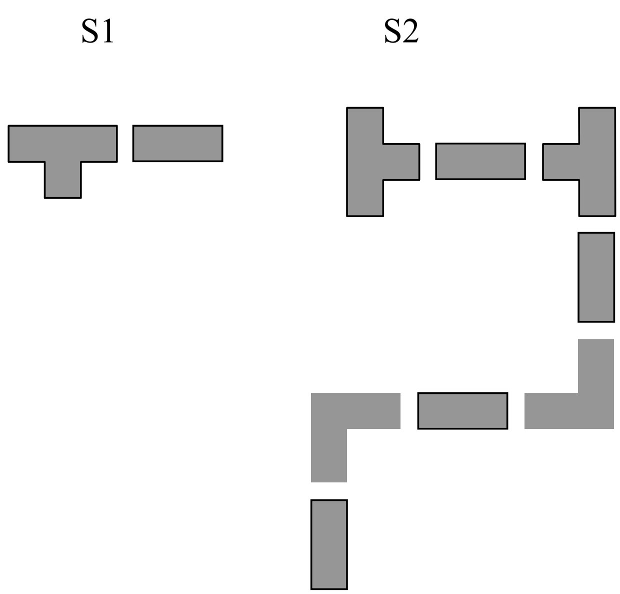


Figure 5.6. S1 Matches S2 by 25%

This metric does not take into account the graph's topology. This is taken handled by the metric for calculating the maximum common subgraph.

5.2.2 MCS (Maximum Common Subgraph) Metric

- σ_{mcs} (S1,S2) is a measure defined by the length of the maximum matching subgraph. If two graphs are nearly identical, σ_{mcs} is close to one. The MCS metric is given by:

$$\sigma_{mcs}(S1, S2) = \frac{\text{length(S')}}{\text{length(S1)}} \cdot \frac{\text{length(S')}}{\text{length(S2)}}$$

where S' is the maximal common subgraph of S1 and S2, that is, the largest graph which is a subgraph of both S1 and S2. The length of a graph is the number of nodes in the graph. For example, consider the graphs *T* and *R* in Figure 5.7. The length *T* is 5, because it has 5 nodes; thus S1=5. Similarly, the length of *R* is 8; thus S2=8. The maximum subgraph is shown on *R* by shading the nodes and arrows on arcs. The MCS is 5 (because 8 is the length of the maximum subgraph). Therefore, using the MCS formula, the MCS is (5/5) * (5/8) = 0.625; that is, a graph match of 63%.

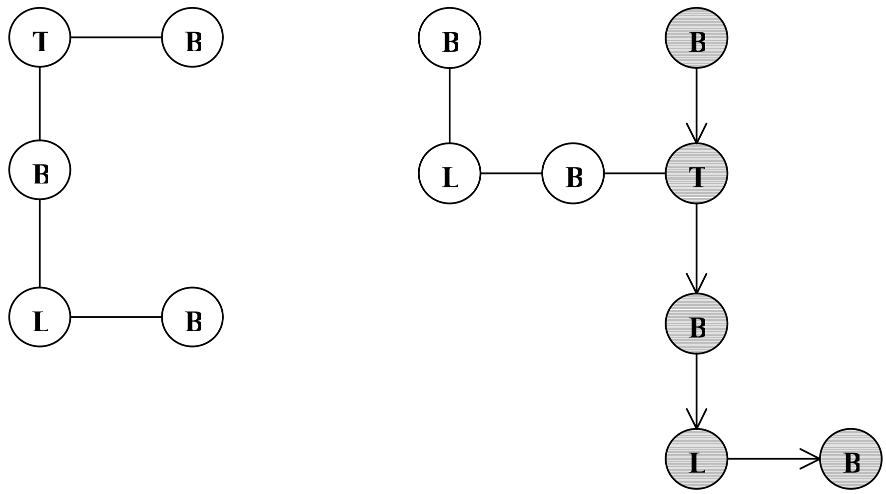


Figure 5.7. Target *T* and Retrieved *R* Graphs (left, right respectively)

For small graphs of up to ten arcs, a search based on direct comparison of all subgraphs of S1 with those of S2 is possible. For larger graphs a strategy based on a preliminary comparison of node types and degree can help to reduce the search time.

5.2.3 Cycle Metric

σ_{cycle} (S1,S2) is based on a count of elementary graph cycles, and is given by the metric:

$$\sigma_{cycle}(S1,S2) = 1 - \frac{|ncycles(S1) - ncycles(S2)|}{max(ncycles(S1), ncycles(S2))} \text{ if}$$
$$max(ncycles(S1),ncycles(S2) > 0, \text{ else } \sigma_{cycle} = 1$$

where ncycles is the number of graphical cycles in S.

Table 5.5 shows σ for a range of graph cycles; for example, when two graphs each have one cycle then these graphs would match by 100%. Similarly, graph G1 with one cycle, and G2 with two cycles would give a match of 50%. When two graphs have zero cycles, then there is a match of 100%.

Ncycles(S1)	Ncycles(S2)	σ
0	0	1
0	1	0
0	2	0
0	3	0
1	1	1
1	2	½
1	3	1/3

Table 5.5 Showing Values of σ for Graph Cycles

5.2.4 Leaf Metric

- σ_{leaf} is based on a count of leaf nodes, and gives the number of branches to a tree. The metric is given by:

$$\sigma_{leaf}(S1,S2) = 1 - \frac{|nleaf(S1) - nleaf(S2)|}{\max(nleaf(S1), nleaf(S2))}$$

if max(nleaf(S1),nleaf(S2) > 0, else $\sigma_{leaf} = 1$

where nleaf is the number of leaves in S. This metric is identical to the cycle metric, and therefore Table 5.5 is again applicable.

The evaluation of the various metrics with respect to different casting design problems is the topic of Chapter 6. In the agenda for future work in Chapter 7 plans are set out to incorporate scale and orientation, which can be treated as further modifications in the light of the results of the test performance graphs.

5.3 Conclusions

Four similarity metrics for shape retrieval were devised and presented in this chapter. These metrics measure the similarity between two shapes and operate on structural graphs. The metrics devised are: Components, MCS (Maximum Common Subgraph), Cycles and Leaves. Justifications why these metrics were chosen were given, including a critique of their strengths and examples of their operation. A generalised similarity metric was presented, which represented a weighted sum of the similarity measures based on different features extracted from two graphs to be compared. The following chapter concerns the evaluation of these shape retrieval metrics to gain a measure of their performance.

Chapter 6 Evaluation of the Shape Retrieval Metrics

This chapter investigates the evaluation of the shape retrieval metrics as devised in Chapter 5 to gain a measure of their performance in solving methoding problems. A *case base*, which consisted of one hundred casting designs was created, with *cases* representing individual casting designs. The cases have been derived from a realistic domain of rotationally symmetric castings and were methoded by an experienced Methods Engineer who worked at the Stone Foundries (London) as a Senior Methods Engineer. A CBR system called *ShapeCBR* was created, which can retrieve the cases for a given *target case*; a target case is a new casting design to be methoded. The operation of the prototype CBR system from a new casting perspective is briefly examined. In the research, the purpose of evaluation is to determine the efficacy of the retrieved cases; that is, the performance of retrieved shapes in predicting the design of a given test case to give solutions for: the moulding direction of the new casting (in other words, its orientation), positions of feeders and chills, and advice. The idea is that given a new casting, T, to be methoded, the similarity metrics (as devised in Chapter 5) are used in a prototype CBR system to retrieve a set of cases $R_{1...n}$. This retrieved set is the solution for methoding the new target case T. The purpose of evaluation is to determine the efficacy of the retrieved set R as methoding solutions. The final stage of evaluation examines the performance of the shape retrieval metrics against human visualisation.

Overview of Chapter

Section 6.1 presents the operation of the CBR system, examining where the cases used in the case base were obtained, how they were represented and how the retrieval process operated. Section 6.2 examines the role of the case base in evaluating the similarity metrics for predicting the design of a given test case. Section 6.3 proposes formulae that give a numerical evaluation to a set of casting problems. Section 6.4 sets up the evaluation experiment by finding the optimised weights defining the total similarity metric as in the first equation presented in Section 5.2 and examines the coverage of the case base. Section 6.5 evaluates the performance of the case base using the optimised weights for predicting the design of a given target case for: orientation, feeders, chills and advice. Section 6.6 examines the performance of the shape retrieval metrics (component, maximum common subgraph (MCS), leaves and cycles) against human visualisation.

6.1 Operation of the CBR System

This section presents the operation of the CBR system for casting design. Following this is an examination of the role of the case base in the evaluation of the similarity metrics.

6.1.1 CBR Procedures

Aadmot and Plaza (1994) describe a generic CBR system as a cyclical process consisting of four parts named the four RE's:

1. RETRIEVE the most similar case or cases.
2. REUSE the information and knowledge in that case to solve the problem.
3. REVISE the proposed solution (if necessary).
4. RETAIN the parts of the problem likely to be useful for future problem solving.

The closest case or cases to the current problem are searched for in the case library (*retrieve*), and the information in the closet case(s) is chosen for use (*reuse*). If the proposed solution fails to solve the problem, the case is *revised*, that is, adapted to suit the needs of the current problem. Once an adequate solution is arrived at, it can be stored for future utilisation, making the system learn from experience (*retain*). The structure of the CBR system used in this research, which conforms to this approach is shown is shown in Figure 6.1.

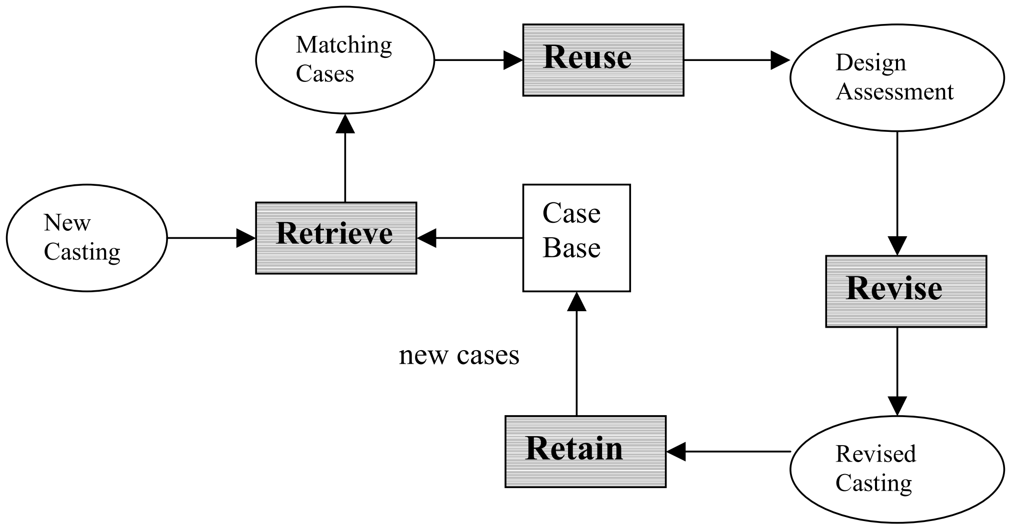


Figure 6.1 CBR Supported Casting Design System (after Lees and Corchado 1999)

6.1.2 CBR Procedures from a New Casting Perspective

The CBR prototype used in this research is called *ShapeCBR*. The system uses as input methoded engineering drawings, which characterise new casting problems. Each problem represents a 'case' and is stored in a case base, which contains the memory of past casting experience. It is envisaged in a commercial system that the Methods Engineer receives drawings from the client in electronic CAD format (see Chapter 7 – Future Work); in the prototype the cases are componentised by hand using a program called *CastAID* (see Paper I in Appendix A for more information about this). Cases contain the drawing slice of a shape (limited to one section slice in the prototype system), and other information about its decomposition; that is, the components that make up the shape and how the components are connected.

The cycle of operations for a new casting proceeds as outlined below:

- First, the system is presented with a new casting (*case*) to be methoded.

- A set of *k* cases is obtained from this current problem by searching the *case base* during the CBR *retrieve* phase, using nearest neighbour matching.

- In the *reuse* phase, the Methods Engineer examines the solutions retrieved, and applies the previous solution in methoding the new case.

- The solution may not be perfect, and the solutions may need to be *revised* (that is, adapted) before they can be applied. Note that in the prototype CBR system used in the research, the process of revision is non-automatic.

- The revised case is then *retained* in the case base. Note that this process has not been implemented in the research CBR prototype.

The following sections expand on this CBR retrieval process.

6.1.3 Case Base

One of the main components of CBR is the case base. This has been defined by Taylor (1997:136) as:

"The memory of past experience."

The case base used in the CBR prototype was populated with one hundred shapes, with the assistance of a consulting Methods Engineer at Greenwich University, who provided access to many paper engineering drawings that he methoded while working as a Senior Methods Engineer over a twenty year period at Stone Foundries in London. The castings used belonged to the domain of rotational symmetry, which included wheels, armatures and cylinders. The domain is coherent from a practical point, in that it can be covered with a limited case base, but furthermore, it is sufficiently varied to encompass a wide range of casting problems.

6.1.3.1 Case Representation

In this research, each case in the case base represents a single shape slice of a casting design. An individual case contains a number of features (*indexes*), which distinguish it from other cases. These features are the CAD drawing of the casting slice and a lower level description in terms of: componentisation, graph structure, cycles and leaves. Cases consist of:

(i) a CAD object
representing an AutoCAD drawing of the shape slice;

(ii) A component decomposition $c_1, c_2, \ldots, c_n$ (where n = number of components)
representing the shape slice as a list of components; that is, Bar, Taper, L-junction, T-Junction, *et cetera*.

(iii) a graph
representing the graphical structure of a shape slice;

(iv) the number of cycles c' (where c' is an integer from 0 upwards)
representing the number of cycles in a shape slice;

(v) the number of leaves L (where L is an integer from 0 upwards)
representing the number of leaf nodes in a shape slice.

6.1.4 CBR Retrieval Process

This is the process used to retrieve cases based on the current methoding problem and involves comparing each case in the case base with a new case, returning a measure of similarity between the new and stored cases.

In Chapter 5, the general similarity measure $\sigma(S1,S2)$ was devised as a measure of similarity between two cases S1 and S2. Pseudo-code for the CBR retrieval procedure is shown in Table 6.1.

```
; given a target case and set of weights, compare all cases in case base to target
; store the target, retrieved case and similarity in global results-table

PROCEDURE CBR-Retrieve(target, case base, weights: leaf, cycle, mcs, comp)
BEGIN
  Case-base-size = number of records in (case base)
  FOR next-case = 1 TO case-base-size
     Results-table(target,next-case,(sim(target,next-case, weights:leaf,cycle,mcs,comp)))
 END
END
```

Table 6.1 CBR Retrieval Process

A program called *ShapeCBR* was developed and used in the evaluation of the case base. More details about the software engineering of the system can be found in Appendix F. The program *ShapeCBR* allows the weights to be manipulated. How the optimum weights were established is presented in Section 6.4.2, with pseudo code in Appendix F. Figure 6.2 shows an example exercise of matching a target case to a retrieved case from the case base. Advice on positions of feeders and chills is annotated on the picture of the retrieved case.

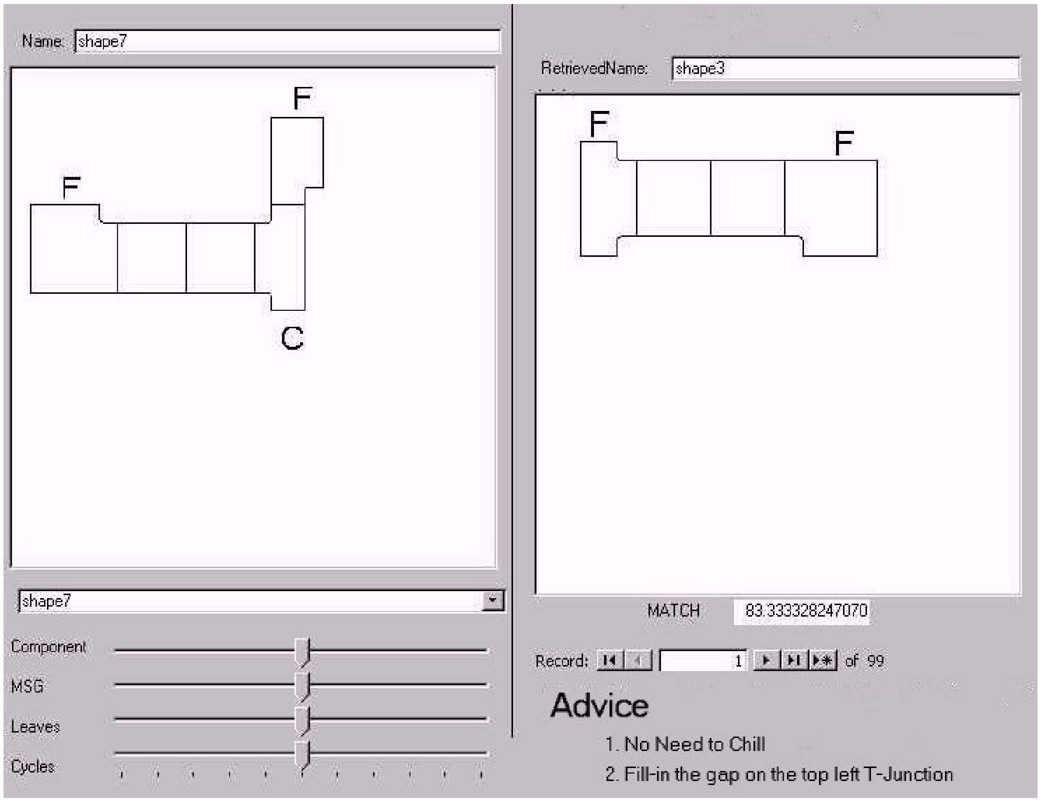


Figure 6.2. Matching a case to a target in the *ShapeCBR* system

Given a target case T, the program returns a set of retrieved cases $R_{1\ldots n}$, from the best to the worst match (R_1 is the top match, R_{100} the bottom). The retrieved set of cases $R_{1\ldots n}$ provide the solution for methoding case T; that is, the methoding solutions S1…S4, for determining:

(S1) The moulding direction (orientation) of the target shape

(S2) The number and positions of feeders in the target shape

(S3) The position of possible chills in the target shape

(S4) Applicable design advice for the target shape

6.2 Role of the Case Base in Evaluating the Similarity Metrics

The purpose of evaluation is about gauging how well the solution set R can be used for methoding target case T; measured by a series of scores for: chills, feeders, orientation and advice. To make this clearer, consider the target case T in Figure 6.3, and the retrieved case R in Figure 6.4. Can case R predict how to method case T? Scores are entered as to how well case R can be used as a solution for methoding T. Low scores indicate that the retrieved case is a poor solution, while high scores indicate that the retrieved case offers a good solution for methoding the target case. High scores imply that the similarity metrics can retrieve shapes that offer good methoding solutions. For this example, the scores entered are:

(S1) 100%: The moulding direction (orientation); can be predicted correctly.

(S2) 75%: The number and positions of feeders; predicts only two feeders out of three.

(S3) 100%: The position of possible chills; predicts correct chill position.

(S4) 100%: Applicable design advice; predicts chill advice assuming advice A1 contains casting knowledge about chilling section lengths.

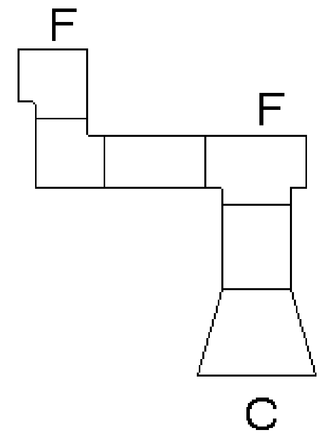


Figure 6.3 Target Case

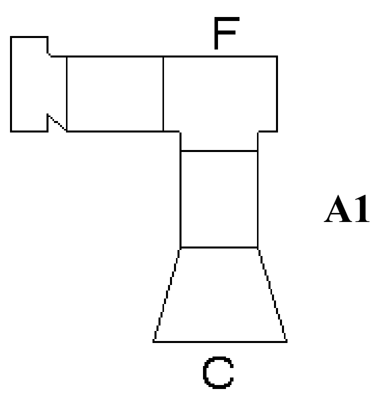


Figure 6.4 Retrieved Case

The scoring was carried out manually, because there is not yet an automatic procedure to achieve this process. The results were stored in an MS Access Table, called *Score Results*. The next section presents the score formulae.

6.3 Formulae for Scoring the Retrieval Solutions

This section gives the formulae that are used in calculating the scores as detailed in Section 6.2. There are four scores: chills, feeders, orientation and advice.

6.3.1 Chill Score

The chill score is:

$$\frac{2*(\text{Matching Chills in Target and Retrieved})}{(\text{number of Target Chills} + \text{Retrieved Chills})}$$

As an example, consider the Target shape in Figure 6.5, the chilled bar; note that the letter C denotes the position of a chill. Assume that the retrieved shape is another bar. Using the chill score formula, the chill score is:

Number of Target Chills = 1
Number of Retrieved Chills = 1
Matching Chills in Target and Retrieved = 1

Chill Score = $\frac{2*(1)}{2}$ = 100%

The number of matching chills in the target and retrieved are combined as one match (not two individual matches). The chill score of 100% means that the retrieved shape correctly predicts where the chills are on the target case; that is the chills are in the same position. Note that the *Matching Chills in Target and Retrieved* is multiplied by two. This is necessary because otherwise the chill score would be 50%.

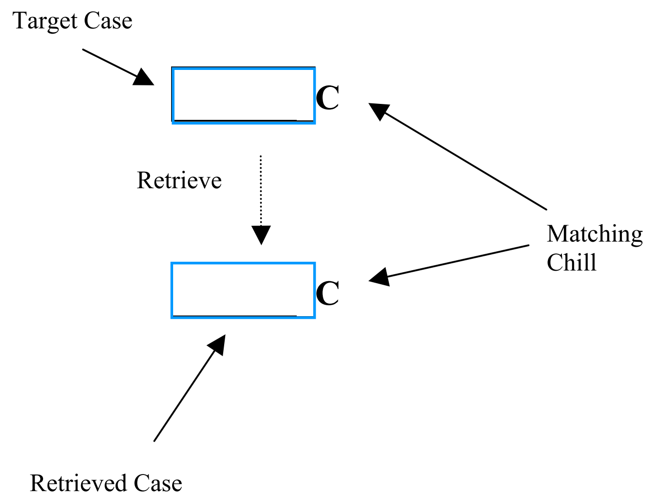


Figure 6.5 Retrieved Case Correctly Predicts Target Chill

6.3.2 Feeder Score

This is similar to the chill score, but feeders are considered instead of chills. The Feeder performance is scored by:

$$\frac{2*(\text{Matching Feeders in Target and Retrieved})}{(\text{number of Target Feeders} + \text{Retrieved Feeders})}$$

As an example, consider the Target shape in Figure 6.6, the fed L shape (note that the letter F denotes feeder positions). Assume the retrieved shape is another L shape. Using the feeder score formula, the feeder score is:

Number of Target Feeders = 2

Number of Retrieved Feeders = 1

Matching Feeders in Target and Retrieved = 1

$$\text{Feeders Score} = \frac{2*(1)}{3} = 66\%$$

The feeder score of 66% means that the retrieved shape predicts 2/3 of feeders in the target case; the result is not 100%, because the target has another feeder (located on the bottom right side of the shape) which has not been predicted. Therefore, in terms of feeders, the retrieved solution is not a perfect solution for methoding the target case.

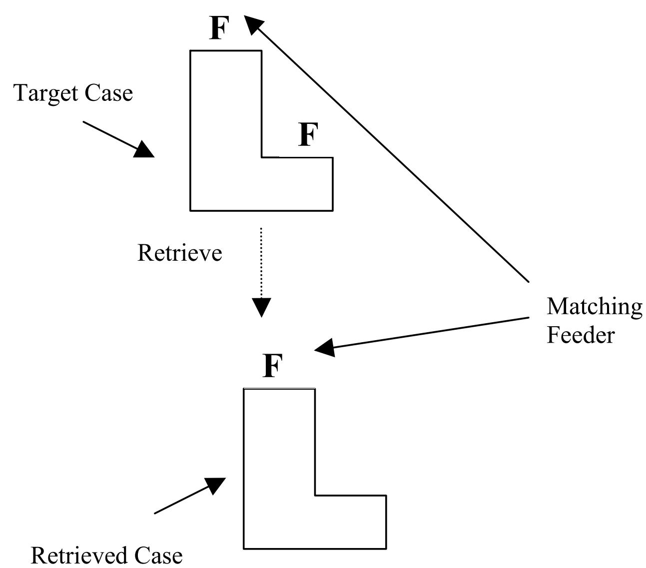


Figure 6.6 Retrieved Case Predicts 66% of Target Feeders

6.3.3 Orientation Score

The moulding direction (that is, the orientation) of a casting is significant, because the heaviest masses of metal should be at the top of the mould to provide direct access for feeder heads (Beeley 1972). A further reason why orientation is important is given by MOD (1979:13) for isolated thick sections. Such sections should not be located at the bottom part of the casting unless feeding of the section can be achieved.

To score orientation, the question needs to be asked: Is it possible to predict the orientation of the target shape from the retrieved shape? This is scored using one of the three values: 100%, 50% and 0%. The meaning for these percentages is shown in Table 6.2.

Score %	**Meaning**
100	The target orientation can be correctly predicted from the retrieved case with 100% certainty
50	It is not possible to predict the orientation of the target case from the retrieved case
0	The orientation of the target is predicted incorrectly; that is, it is the wrong way up

Table 6.2 Orientation Scores

6.3.4 Advice Score

Each case has advice relating to its design. Advice falls into two categories:

1. General Advice; for example, “fill gaps in top-left T-junction”.
2. Re-Design Advice, where casting problems have been identified (such as thin sections), and re-design is suggested.

Within these two categories, advice can be of 4 types: $\S_{1\ldots4}$

$\S_1$ Advice in the retrieved case is not applicable to the target case.

$\S_2$ Advice in the retrieved case is applicable to the target case, but cannot be applied, because it would inappropriate to apply the advice.

$\S_3$ Advice in the retrieved case is applicable and can be applied to the target case.

$\S_4$ The target case may have advice not covered by the retrieved case; that is, missing advice.

The formula used to score advice is:

$$\text{Advice} = \frac{\S 3}{(\S 1 + \S 2 + \S 3) - \S 4}$$

As an example, consider Figures 6.7 and 6.8, which show a target case and a retrieved case respectively.

Given the target case and the retrieved case, the percentage of applicable advice for the target is:

$\S_1$ Advice in the retrieved case is not applicable to the target case = 0

$\S_2$ Advice in the retrieved case is applicable to the target case, but cannot be applied, because it would inappropriate to apply the advice = 1. This is because the retrieved case gives advice to chill legs, but the left leg of the target case does not need to be chilled.

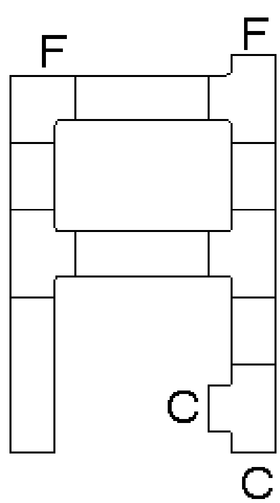
§3 Advice in the retrieved case is applicable and can be applied to the target case = 2

§4 The target case may have advice not covered by the retrieved case; that is, missing advice = 0 since there is no missing advice.

$$\text{Advice} = \frac{2}{(0+1+2)-0}$$

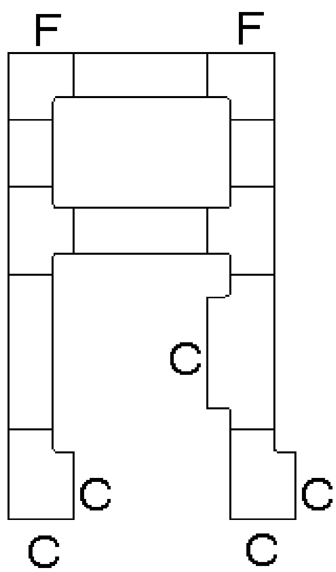
$$= 66\%$$

The retrieved case has advice that is 66% applicable to the target case. The higher the score, the better the similarity metrics are in retrieving cases with good methoding advice.



Advice
1. Thicken top left and right sides, within cycle
2. If not thicken, then possibly taper top right hand side

Figure 6.7 Target Case



Advice
1. Thicken top left and right sides, within cycle
2. If not thicken, then possibly taper top right hand side
3. Always chill legs

Figure 6.8 Retrieved Case

6.4 Evaluation of Shape Retrieval

In this research the purpose of evaluation is to measure the efficacy of the shape retrieval metrics in retrieving castings that offer *good* methoding solutions; that is, the retrieved shapes should predict how a target shape should be methoded. The evaluation measures this performance.

Before the metrics could be evaluated, it was necessary to set up the evaluation experiment. This involved entering scores (the score formula were detailed in Section 6.3) for each target case and a set of retrieved cases in the case base using equal weights. Following this, an iterative process found the optimum weights.

The first stage of evaluation explored the scored results for: orientation, feeders, chills and advice.

The second stage of evaluation explored the results of the individual metrics against the human visual match.

Further information is given in Chapter 7 regarding the opinions of a distinguished world expert on casting on the performance results of the evaluation.

6.4.1 Experiment Set Up

A series of score formulae were presented in Section 6.3. Before evaluation could begin it was necessary to score the retrieval set for each target case in the case base. It proved impracticable to enter scores for all one hundred cases in the case base, because this meant that scores would have to be entered for ten thousand (100*100) records. Therefore, a pragmatic decision was made to enter scores for the top five retrieved cases, instead of all 100 retrieved cases.

The metric weights (mcs, leaf, comp, cycle) were set to an equal value (0.5). Scores were then entered for the 100 cases in the case base to a retrieval depth of five; that is, scores were entered only for the top five retrieved cases for practical reasons as mentioned in the previous paragraph. These scored results were stored in a MS Access table called Score Results; this table had the fields: *TargetCase, Retrieved Case, Orientation, Feeders, Chills, Advice*. After the scores were entered, *Score Results* contained 500 records (that is, 100 cases * 5 retrieved cases).

6.4.2 Determine Optimal Weight Settings

An iterative process found the optimal weights, and the pseudo-code to do this is presented in Appendix F. The process works by running through all the possible weight combinations from zero to one using a step interval of 0.2. For all the target cases and a particular weight setting, the best matching cases were retrieved, along with the scores for feeders, chills, advice and orientation (extracted from Score Results table; see Section 6.4.1.) These scores were averaged over the case base as a test set, using the frequently used technique of removing each case in turn from the case base and using it as a test target case and storing the results (with the weight settings) in a table called Human Performance Comparison. The weights were then set to another iteration and the process repeated. Finally, when the iteration ended, the Human Comparison Performance table was searched for the best average scores. The record with the highest scores thus had the optimal weight settings. After the iterative process ended, the optimal weights were as in Table 6.3.

a) components (σ_{comp})	0.6
b) maximum common subgraph (σ_{mcs})	0.8
c) cycles (σ_{cycles})	0.4
e) leaves (σ_{leaves})	0.2

Table 6.3 Optimal Weights

This seems to indicate that all of the similarity metrics were helpful, but the MCS and component metrics were the most important, with leaf metric the least significant metric. Table 6.4 shows the performance on the case base (i) with all the weights set equally to the value 0.5, and (ii) with the final optimised weights.

	(i) Equal weights	(ii) Optimised weights
Orientation	75.50%	79.50%
Feeders	74.28%	76.08%
Chill positions	57.28%	59.27%
Advice	82.50%	84.50%

Table 6.4 Performance for the whole Case base

The performance figures in Table 6.4 can serve as a benchmark for future maintenance of the case base, since future versions of the case base should perform at least as well on the original 100 cases as the performance figures in Table 6.4. This requirement can also help guide decisions on inclusion or deletion of cases.

6.4.3 Case Base Coverage

The coverage of the case base, that is, how the cases in a case base are spread over the problem space, is important (Smythe and McKenna 1998). For instance, a target problem is solved with a high degree of accuracy from regions of high density, but likely to be unsolvable from sparse regions.

The approach taken in this research for considering the distribution properties of the case base is to represent cases as points in the multidimensional space of features as given in Chapter 5 – Section 5.1. All of these features give rise to meaningful dimensional integer constants, and the space is closely connected with the similarity measure employed in retrieval. Distribution can be assessed visually by examining 2-D plots of cases on pairs of features. However, this is difficult to assess in view of the large number of pair-wise plots necessary for the complete space. A better view can be obtained by looking at the principal components of the space (Jackson 1991), as shown in Figure 6.9, which has the advantage of combining all the features. The plot shows a visualisation of the case base with equal weights and it gives the impression of a single coverage group of cases, contained within a convex boundary. Within the boundary there is evidence of gaps and areas of high density. The distribution of a set of 20 test cases can also be seen on this plot, and seem to provide a representative spread.

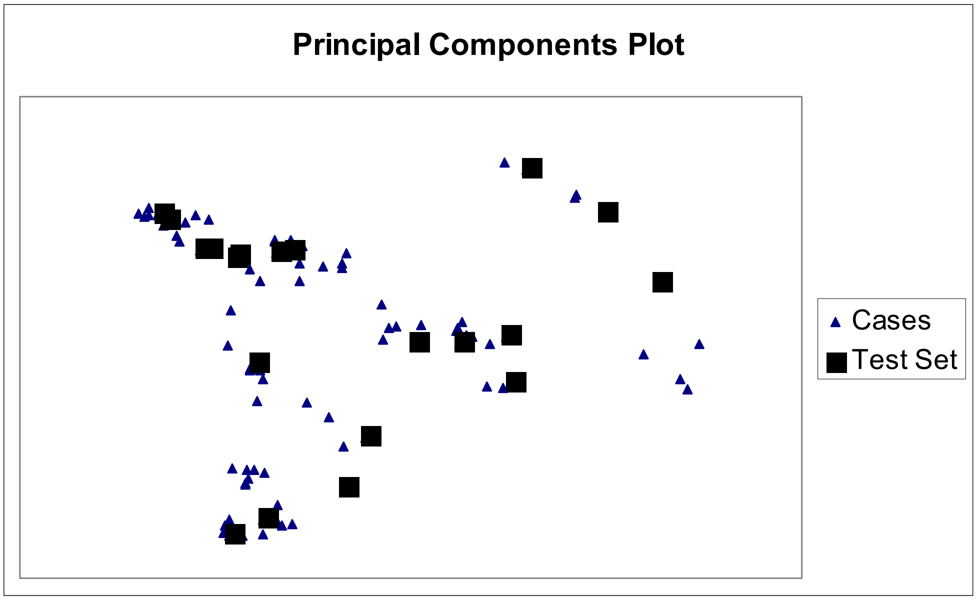


Figure 6.9 Principal Components Plot Showing Distribution of Cases and Test Set

6.5 Performance of the Case Base

This section presents results for the overall performance of the case base using the optimised weights for the following measures: orientation, feeders, chills and advice. A trial set of 20 target cases was taken, as shown in Figure 6.9, and a number of randomly selected case bases of a given size were used to predict these cases. Average scores for the performance of case bases of each size were plotted against case base size. Results are shown for the optimised similarity metric:

$$\sigma = (0.2\sigma_{leaves} + 0.6\sigma_{comp} + 0.8\,\sigma_{mcs} + 0.4\sigma_{cycles})/4$$

6.5.1 Orientation Performance

Figure 6.10 shows the orientation performance. The graph shows a steady growth in performance that appears to be flattening out at around 80%. This seems to indicate two important facts. Firstly, that there is not much to be gained by putting in many more cases. Secondly, therefore, the 80% efficiency must be a measure of the efficiency of the retrieval process itself, since it can't be improved by modifying the case base.

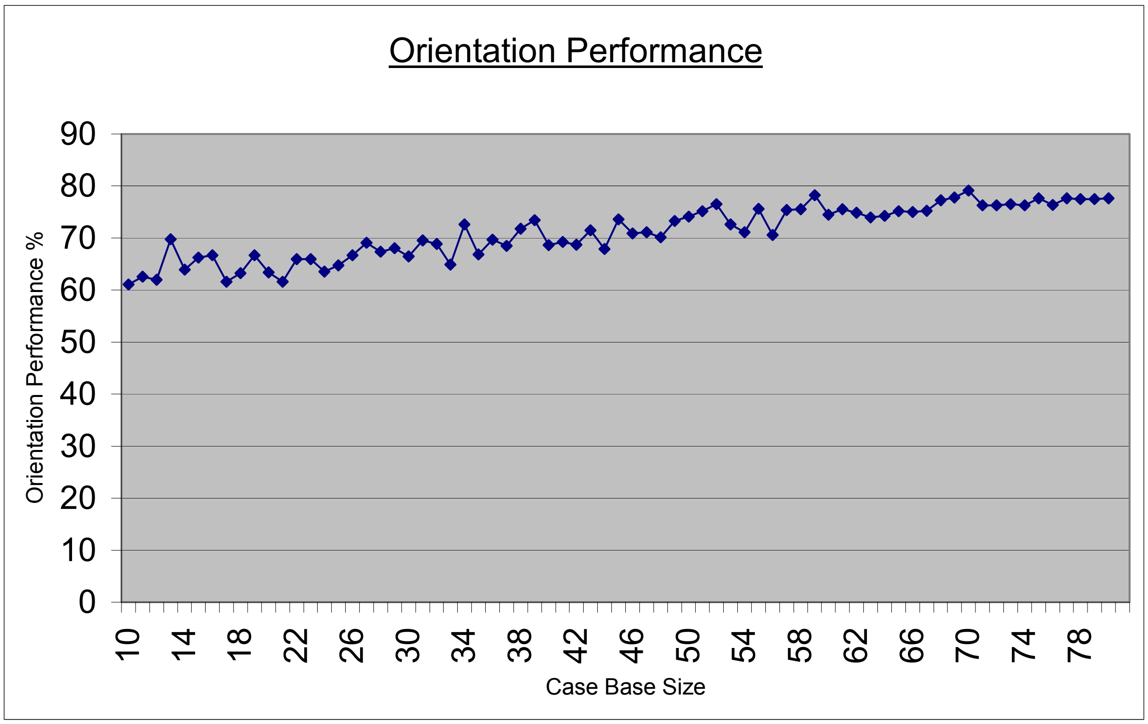


Figure 6.10 Orientation Performance

6.5.2 Feeder Performance

Feeder performance is shown in Figure 6.11. It shows that as the case base increases, so does the feeder performance, levelling out at around 70%. Again, the results indicate that there is not much to be gained by adding further cases.

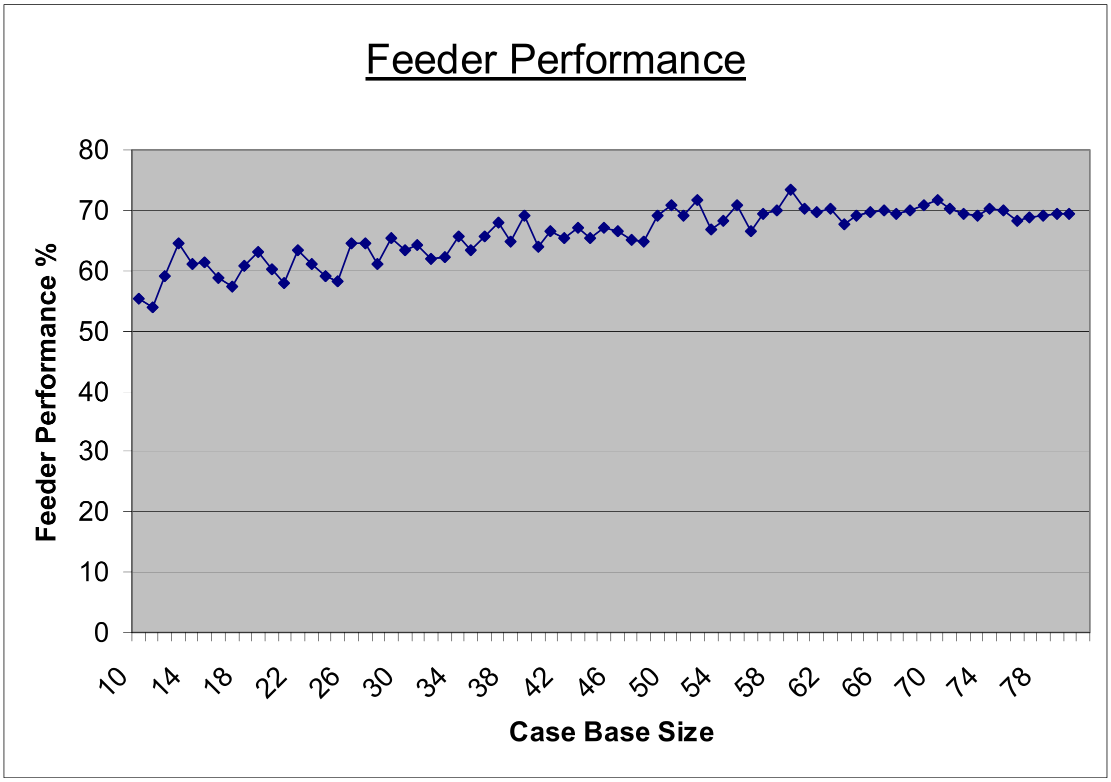


Figure 6.11 Feeder Performance

6.5.3 Chill Performance

Figure 6.12 shows the results for chills, a most disappointing result. A strong reason for this low performance is coupled with the way feeders can be substituted for chills. Indeed, there are empirical rules and knowledge to assist whether chills or extra feeders should be used. Chills, which are used to make the molten metal freeze more quickly, are used to minimise excess feeders and thus lower the cost of casting. The plate in Figure 6.13 could be fed using two feeders, but it is more cost effective to use one feeder and one chill. However, using the chill metric, the chill score of Shape 2 comes out at 0%. A way forward is to develop more sophisticated metrics, which use chill substitution knowledge. Extensions to the metrics for optimising shape retrieval are investigated in Chapter 7 - Conclusions.

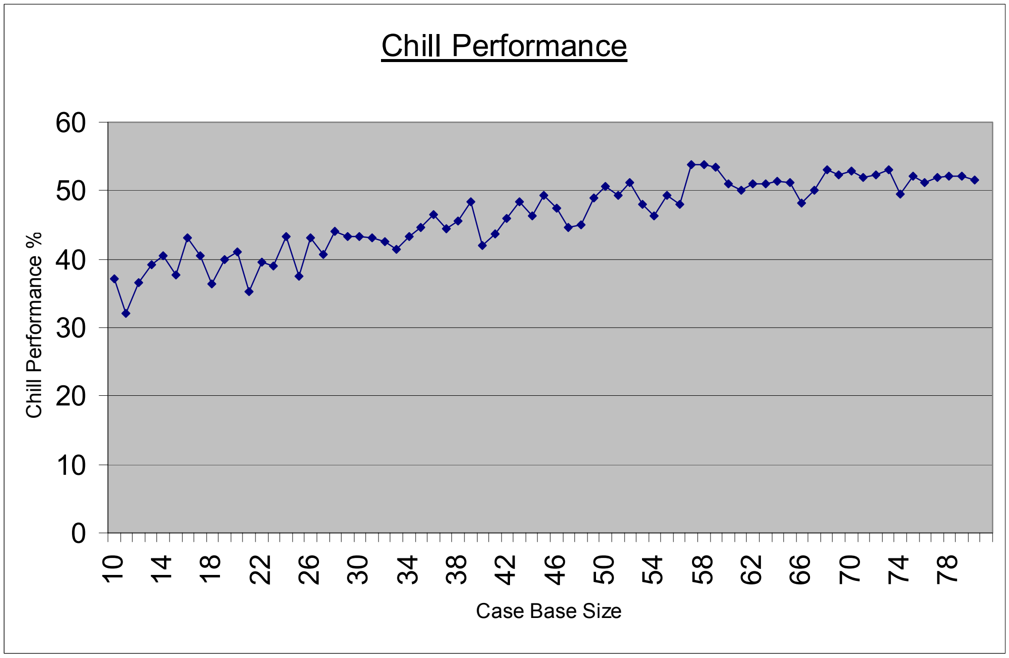


Figure 6.12 Chill Performance

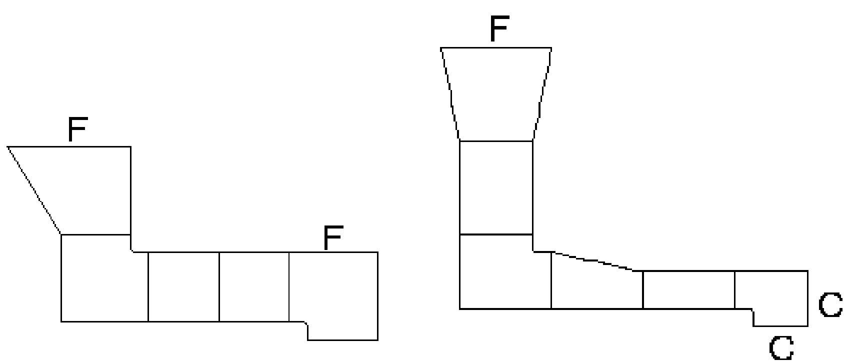


Figure 6.13 Feeder Substitution of Shape 1 into Shape 2 (left and right)

6.5.5 Advice Performance

As the case base increases, so should the advice. Figure 6.14 show that the correct advice increases up to 50 cases then levels off after peaking at 75%. The implication here is that there is no point entering many cases with similar advice as this won't make much difference to advice performance. Rather, it suggests that it may be better to have a policy of entering essential cases that hold the most significant advice.

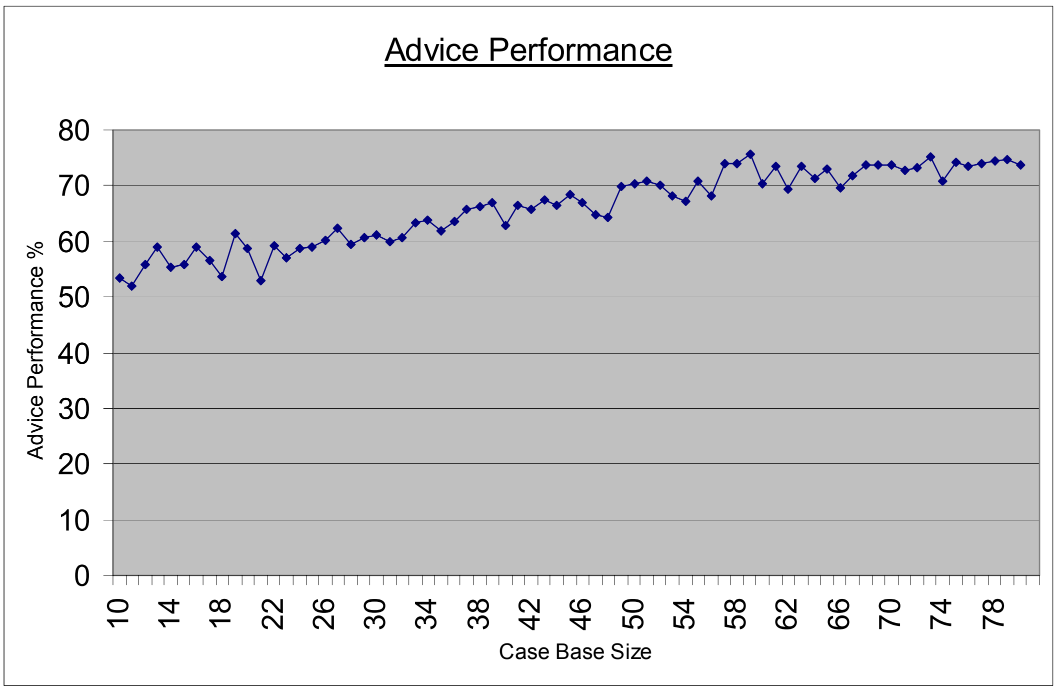


Figure 6.14 Advice Performance

6.6 Performance of Shape Retrieval Metrics against Human Visualisation

A human visual match was associated for each target case and set of retrieved cases. This match was scored by asking: "How much does the retrieved shape look like the target case?" A score, 0 to 100% was entered. In this section, this human visualisation score is analysed against the shape retrieval metric performance for: components, MCS (maximum common subgraph), cycles, and leaves. Scatter plots for each metric are shown and these graphs have the human visualisation match on the horizontal axis in increments of five, and the average of the particular similarity measures along the vertical axis. So for example, the point (100, 96) means that for all those comparisons with the human visual score 100%, the average weighted similarity score was 96%. The goal anticipated is for each of the graphs to be a smoothly increasing curve, and not a straight line, because this matches the performance of the human expert. Finally, results are given for the performance of the combined metric.

6.6.1 Components

Figure 6.10 shows the component performance. This graph shows that there is an approximate correlation to the human visual match. However, there is much individual disagreement between the human visual and component count match, and a major reason for this is as a result of the metric not considering the spatial layout of the shape; that is, two shapes may have the same components, yet diverge entirely when considering the component connectivity. Nevertheless, from the graph of Figure 6.15 it is evident that this simplistic metric scores unexpectedly well; for example, the points (81, 81) and (100,96) which correspond closely to human expertise. The main reasons for these high scores is that particular classes of castings are made up of particular components that are more or less unique to that class of shape; for example, there are some shapes that have more cross-junctions, while others cases have more T-junctions, and these shapes can be grouped as belonging to a particular class of casting.

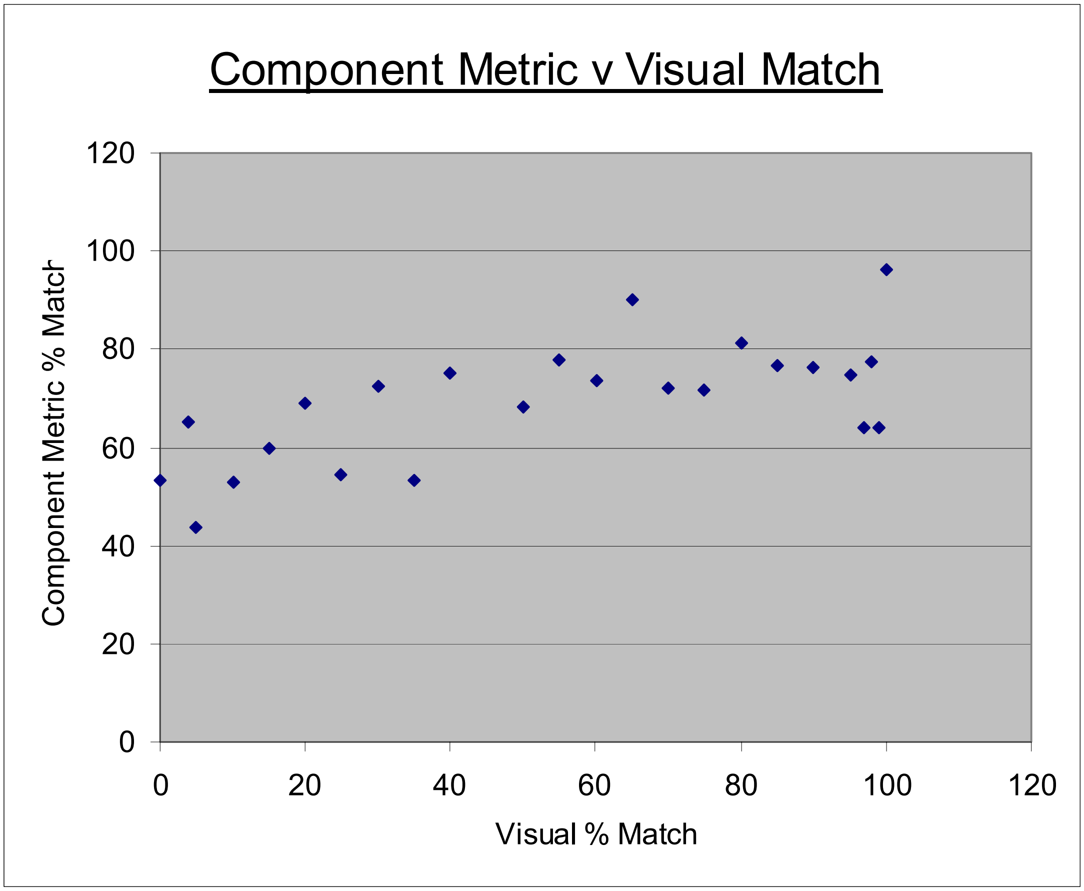


Figure 6.15 Component Performance

6.6.2 Maximum Common Subgraph (MCS)

The performance of the MCS, shown in Figure 6.16, increases steadily, matching the human visual match fairly well; for example, the points (100, 94), (40, 43). However, there are some low points on the graph that need further investigation; for example, the points (70, 41) and (35,10). The outlier (35,10) can serve as a focal point to show how the MCS algorithm could be refined. Although the visual match is high (35%), the MCS is low (10%). The reason for this is that a T-junction in the middle of the matching group is substituted by an L-junction. Figure 6.17 demonstrates this with two shapes (6.17a and 6.17b respectively) that are for casting purposes practically identical. Yet, it would be better if the MCS algorithm matched the L-junction in shape 6.17b as being closer to the T-junction in shape 6.17a, since only the right top bar is missing. Hence a more sophisticated MCS algorithm, taking account of *nearly* matching components may perform more successfully. Furthermore, since the metric does not consider the orientation of components, two shapes that may have an identical spatial layout, yet are oriented differently and this would give an identical match. For these reasons, the MCS match does not perform at a human level of expertise.

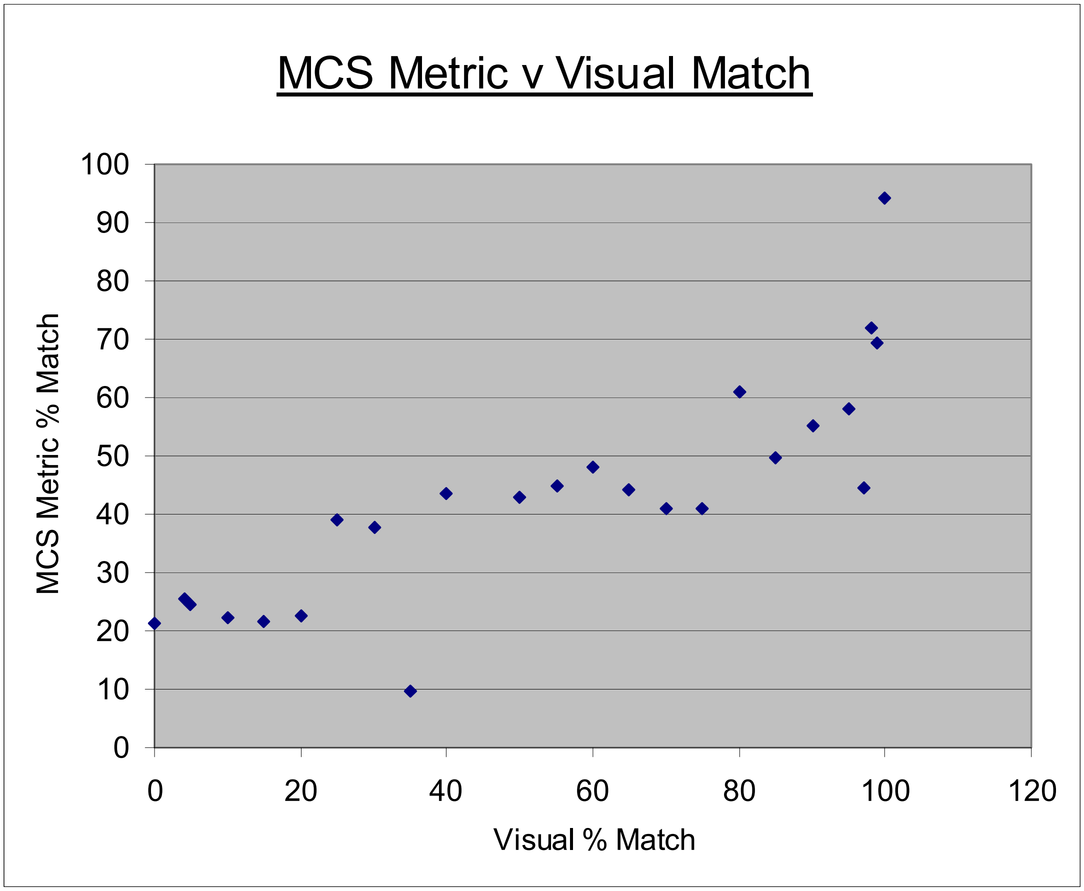


Figure 6.16 MCS Performance

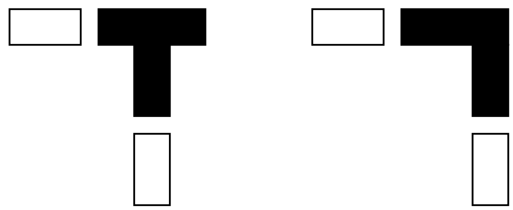


Figure 6.17. Shapes 6.17a and 6.17b

6.6.3 Cycles

Figure 6.18 shows the results for cycles. The fact that the graph has a line of points along the 100% cycle level results from the cycle metric giving a score of 100% when two graphs each have zero cycles (as discussed in Section 5.2 - Generalising the Similarity Metrics). Therefore, regardless of the visual match, there was 100% cycle average. It might be argued that two graphs with no cycles should have been given a match of 0%. However, very few cases in our case base had cycles, so the graph would instead have had a line of points along the 0% level. These results indicate that cycles are not an important feature for shape retrieval. However, this is most likely to be because of the limited number of test cases with less than one cycle. Further work can investigate further tests with cases with having more than two cycles to gain a better measure of cycle efficacy.

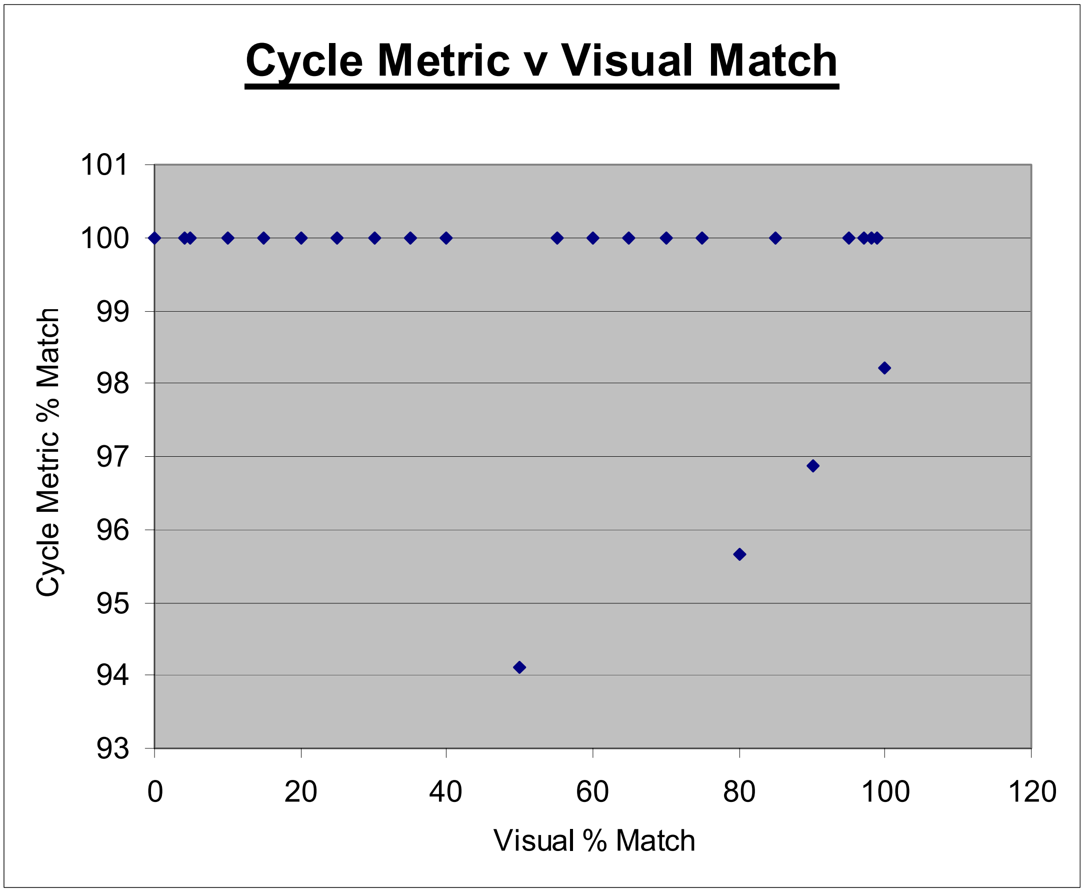


Figure 6.18 Cycle Performance

6.6.4 Leaves

Figure 6.19 shows the visual match for leaves. There appears to be an indeterminate correlation between the leaf metric and human visual match. A rationalisation for this result is that although two shapes might have identical leaves the structure of these shapes might differ greatly. Consider for example, two shapes having identical leaf nodes: L-junction and Bar; thus giving a match of 100%. Although these shapes give an identical match it still does not impart anything important about the *overall* shape structure in a way, for instance, that the maximum common subgraph does. Indeed, by re-examining the case base it was found that although shapes may have identical leaf nodes they would give low visual match scores, a result confirmed in the graph of Figure 6.19. Therefore, the leaf node is an insignificant metric for shape retrieval.

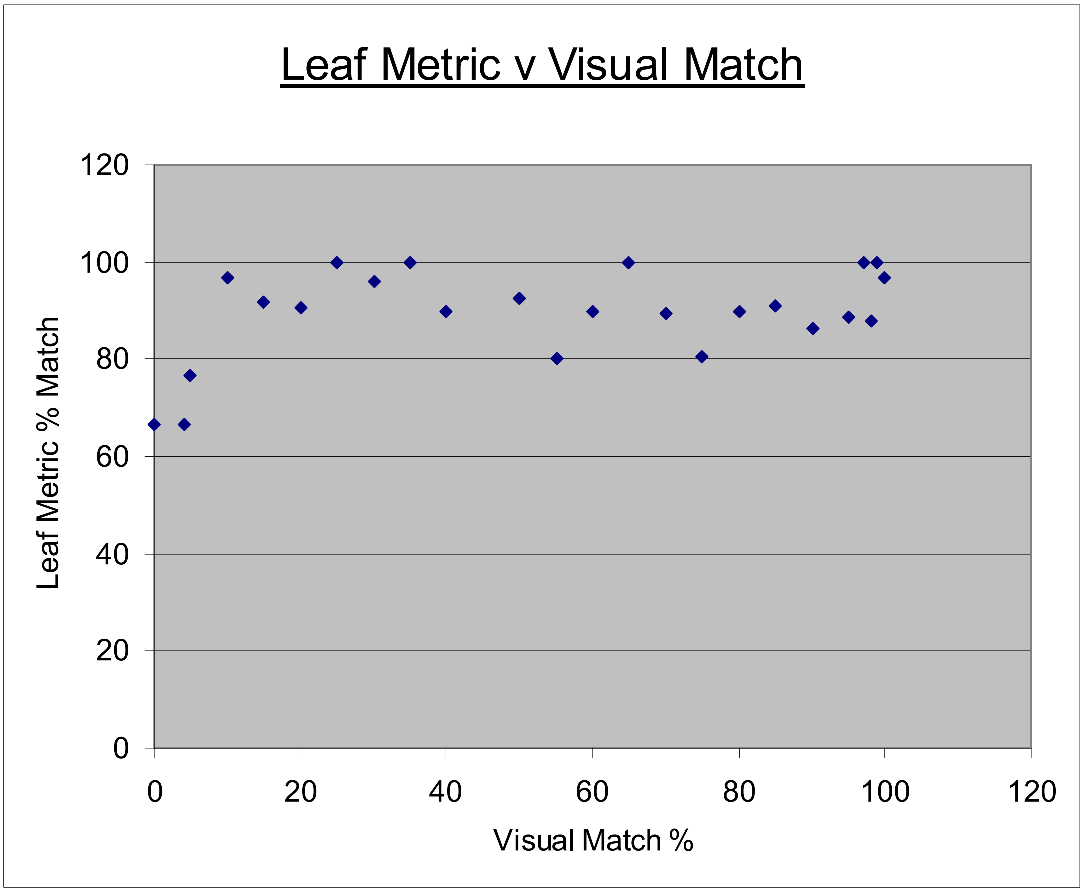


Figure 6.19 Leaf Performance

6.6.5 Visual Versus Combined Scores

Figure 6.20 gives a plot of the human visual match against the combined metric of:

$$\sigma = (0.2\sigma_{leaves} + 0.6\sigma_{comp} + 0.8\,\sigma_{mcs} + 0.4\sigma_{cycles})/4$$

If the metrics are to match human retrieval then the graph should show a steady increase, implying that there is a fairly good correspondence between the weighted similarity and human visual matching. Figure 6.20 is interesting in this respect, showing that the weighted similarity is better than any of the individual matches.

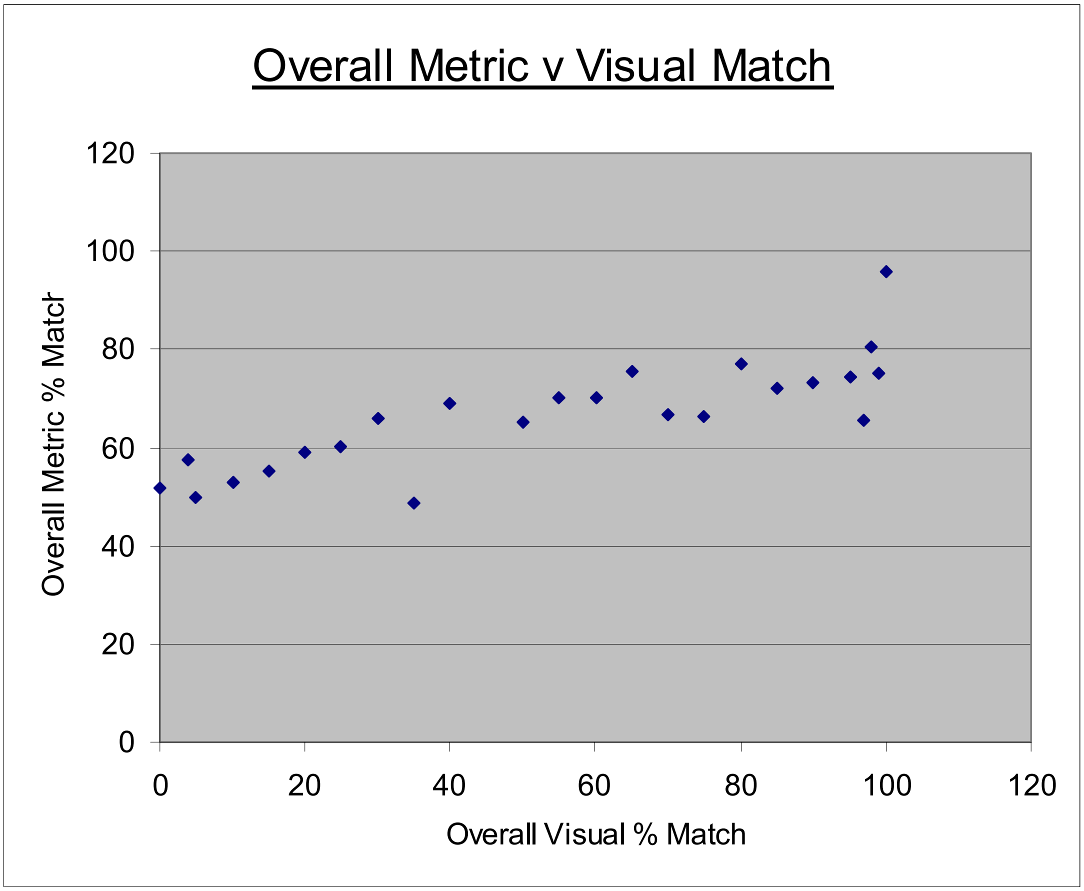


Figure 6.20 Total Case Performance

However, there are some anomalies. From Figure 6.20 it is possible to identify these evident errors; for example, the obvious outlier at (35,50). Investigation of these shows clearly the weaknesses in the metrics. The problem comes down to two inadequacies, both of which can be corrected:

1. Insufficiency of orientation information when considering two components being joined together. For example, a right-hand L and a left-hand L are treated just as an L at present.

2. Insufficiency of matching on components themselves. For example, a long thin T is the same as a short fat T at present.

3. No match on analogous components, e.g. a Bar is similar to a Taper. A T-Junction is similar to an L-Junction.

It is probable that the inclusion of these retrieval modifications may bring retrieval closer to human retrieval. Further improvements to the metrics are examined in the following chapter in an agenda for further research.

6.7 Conclusions

The evaluation focused on the performance of retrieved shapes in predicting the design of a given test case to give solutions for: the correct orientation, locations of feeders and chills, and advice.

Initially, the operation of the CBR system for casting design was presented. An experienced Methods Engineer acted as a consultant at Greenwich University, and provided over 100 castings, which formed the basis of the case base; these cases were obtained from a real casting domain of rotationally symmetric shapes. A CBR system (Shape*CBR*) was created for the evaluation of the case base. The coverage of the case base, that is, how the cases in a case base are spread over the problem space is important and by examining the case base using the principal components of the space the distribution was revealed showing that there was a representative spread.

The overall performance of the case base with a set of optimised weights was in the range of 59.27% to 84.5%. Following this the performance for orientation, feeder, chill and advice was examined. Orientation performance increases as the case base increases. Feeder performance improves as the case base increases in size, although from approximately sixty cases onwards there are not considerable improvements in feeder performance; thus there is not much to be gained by adding more cases. The chill performance of 59.27% was a disappointment, and the main reason for this low performance is associated with chill-feeder substitution, in which feeders can be used instead of chills. The performance of advice showed that it increases as the case base increases; however there is not much of a performance increase after fifty cases and this suggests that it is more practical to enter key cases with the most significant advice.

Finally, the performance of shape retrieval metrics against the human visual match was examined. The main conclusion was that the size of the maximum common subgraph, and the number and type of components were the most important metrics. The cycle metric was found to have little effect on shape retrieval, and the leaf metric was found insignificant for shape retrieval.

Chapter 7 Conclusions

This chapter concludes the thesis with a summary of work carried out in answering the research questions posed in Chapter 1, followed by the contributions of the research and new questions raised by the research, which sets an agenda for future work.

7.1 Summary of the Thesis

The crux of the research has been concerned with a central question, asking whether it is possible to build a CBR retrieval system that performs nearly as well as a human expert in retrieving analogous casting design cases in the problem domain of rotationally symmetric castings. Initial knowledge elicitation suggested that a representation could be based on a shape componentisation scheme that is specific to Methods Engineering corresponding to the way Methods Engineers reason about solid objects. A novel contribution of the research found that Methods Engineers mentally decompose 3-D casting shapes into an assemblage of 2-D vertical and horizontal cross-sections, with each 2-D casting section made up of specific components of structural elements (such as cross, T and L) and connecting elements (such as bar and taper). The research examined the possibility that an approach based on this idea of using cross-sections could provide near-expert retrieval. Once it was established that a representation based on an 'natural' decomposition scheme existed, a research programme was set out, its objective to construct an automated retrieval system, and to judge how well it performed against human expertise.

At the international foundry conferences: CastCon'97 (Preddy, Knight, Cowell, Mileman 1997) and the 34th Foundry Days (Mileman 1997) there was much interest from the audience in the approach of casting design using a CBR approach based on the Methods Engineers shape decomposition as presented in this thesis. In both these conferences, the verdict from the scientific community and practising Methods Engineers was that the approach is both a valid and an interesting way for casting design.

Various performance measures were devised to quantify how well human retrieval and automatic retrieval operate. A trial domain consisting of one hundred shapes was constructed with the assistance and guidance of an experienced Methods Engineer who acted as a consultant at Greenwich University. A number of realistic problems that the system should provide answers to were chosen, these being:

(I) Correct orientation
(II) The number and positions of feeders
(III) The position of possible chills
(IV) The need for chills
(V) Special problems encountered with this shape (for example: "bar too thin", "fill-in top gaps").

Comparisons were made using different similarity metrics for the retrieval process. Four similarity metrics were devised and the metrics extracted features from a shape. These indexed features were:

1. The types of components that make up a shape
2. The connectivity of components (Maximum Common Subgraph)
3. Types of leaf nodes
4. Graphical Cycles

These were combined into a general similarity metric that returned a weighted sum of the shape similarity (one being a perfect match, zero no match). Evaluation showed that the computerised metrics approached human retrieval in the range 59.27% to 84.5%. Shape retrieval could be measured without their needing the input of Methods Engineers, because the scoring formula did not need their expertise and so they were not involved in the evaluation stage. However, the final results were discussed and shown to the distinguished professor of casting, John Campbell, who has looked at the results of this research. He has remarked that the findings are very encouraging (Campbell 2001). This has given further credibility to the research and rise to an exciting prospect that the research prototype could be extended into a commercial system that practising Methods Engineers might one day use.

7.2 Contributions to Knowledge of the Research

The contributions of this thesis in answering the research questions posed in the introduction in Chapter 1 are outlined below.

The research contributes a novel CBR process for shape retrieval in the casting industry, with pointers for the application of the method for the wider domain of shapes. The first major contribution identified in the knowledge elicitation stage was a 'natural' shape componentisation scheme of 2-D section slices into elementary components. Previous approaches used arbitrary divisions, whereas this research focused on a human-based mechanism for shape representation and retrieval. Novel contributions have been made in identifying, abstracting and formulating this process, and its evaluation.

A series of similarity metrics for shapes in the casting domain were proposed. These included both metrics based on feature-value pairs, and similarity based on graph matching. Although algorithmically simple, a significant contribution of this research was the success of using such rudimentary metrics for shape retrieval, yet still approaching human expertise.

Using a prototype CBR retrieval system a benchmarking framework for measuring the feasibility of the approach presented in this thesis was carried out. In evaluation, the computer representation of shapes based on the one used by Methods Engineers has been shown to be capable of being successfully applied and used to reason about a family of rotationally symmetric shapes. The work is a major contribution of applying CBR to the casting industry.

7.3 Future Work

Although the contributions to knowledge of this research as stated above are significant, another real contribution of this work is the research agenda that it motivates. The main aim of this agenda is summarised, followed by examining improvements to the current work to advance the efficacy of shape retrieval.

7.3.1 Development of Further Metrics

A series of new metrics may improve retrieval performance, particularly if component parameters and orientation are included in the general similarity metric. Examination of the performance results of the prototype system indicated that due to absence of orientation information several false matches were being made. This indicates that the addition of orientation could improve performance.

In Chapter 4, the importance of extra information on the orientation of components was briefly investigated, and an enhancement to the graphical representation proposed in the form of a bearing attribute for each junction section. The MCS metric could be improved by taking account of the orientation information in the enhanced graph structure. However, a problem arises, because it may not be sufficient for two joined components to be of the same type in each of the graphs, and components that are in the same orientation. This tends to reduce the size of the MCS for differently oriented structures. It should be noted that the MCS algorithm needs to calculate an arbitrary initial angle when making the first match between components of the same type. This is because there is an arbitrary angle specifying ‘north’ for sections perpendicular to the axis of rotation, and a 180 degree indeterminacy of the direction of the axis of rotation for sections including the axis of rotation. In each case, this initial angle can be calculated by determining that the first components of the same type are to have the same bearing.

The set of components presented in this research have bearings of 0^o, 90^o, 180^o, 270^o. If curved joining sections are not used then there is no problem in judging whether two components have the ‘same’ orientation. However, the matching of bearing in the general case presents a problem if components are generalised other than the right-angle components used in the research. If two bearings are the same then the algorithm can treat the two components as having the same orientation. However, if they are not exactly the same a question arises concerning the level of tolerance allowed. Two ways that this problem could be solved are presented below:

1. Allow a measure of similarity of orientation between two components, such as:

$$\mathbf{d_{12} = 1 - (\Phi_1 - \Phi_2)/90 \,, \quad |\Phi_1 - \Phi_2| < 90}$$
$$\mathbf{d_{12} = 0 \,, \qquad\qquad\qquad\quad |\Phi_1 - \Phi_2| > 90}$$

By applying this to all the components in the MCS found by ignoring bearing information, a measure can be obtained as to how well the two MCSs are oriented by:

$$\mathbf{D = \Sigma_{junctions}\, d_{ij} / (number\ of\ junctions)}$$

A search for the maximum of some joint statistic could be carried out representing: Length of MCS * Orientation of MCS.

The second approach might be to have a "cut-off" angle or tolerance beneath which could be used to determine whether two components are oriented similarly. For example, two components are oriented similarly if:

$$|\Phi_1 - \Phi_2| < 5^o$$

This would allow the MCS algorithm to proceed as long as two components are roughly in the same direction.

2. The next approach could consider the simple count of component types. Bearing information can be useful here in weeding out dissimilarly oriented shapes. For example, Figure 7.1 shows two radically different shapes with the same L-component counts. However, the counts of oriented L-components show a zero match.

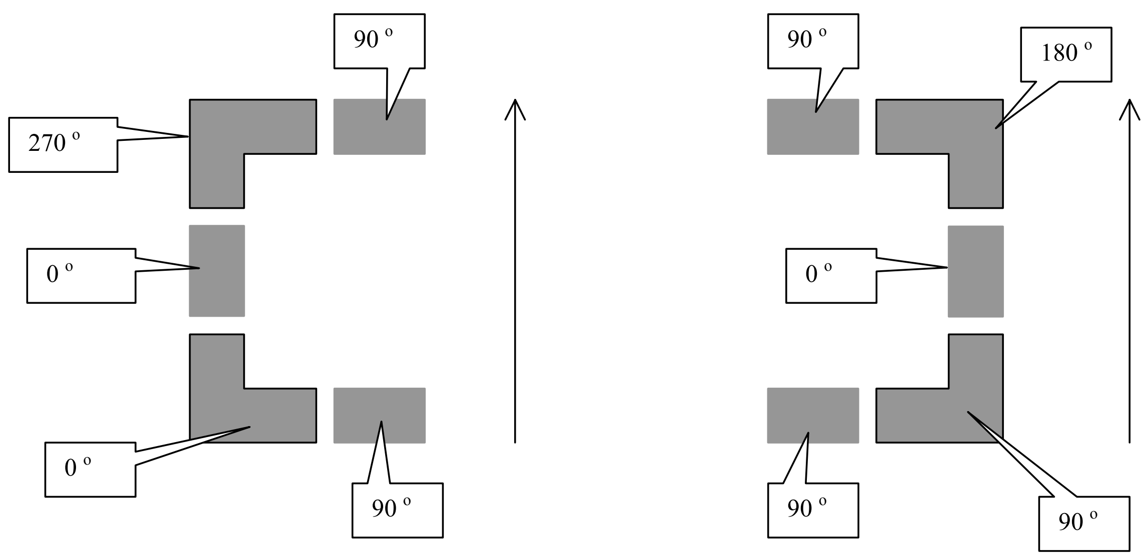


Figure 7.1 Different Shapes, but same L-Component Counts

For this measure, it would also be an advantage to add bearing information to the joining components. Since there is naturally a 180 degree indeterminancy in any bearing information, this information can be restricted to the range: 0^o, 180^o. Figure 7.1 also shows bearing information on joining components.

7.3.2 Matching More than One Section Slice

It is assumed in this research that each shape has only one section slice, whereas the shape data model presented in Chapter 4 - Section 4.5 allows an arbitrary number of slices of horizontal and vertical orientations. Future work should focus on developing the *ShapeCBR* prototype to allow any number of slices for a given shape. A slice matching metric could be developed, utilising the general similarity metric already developed in the research (see Chapter 5 - Section 5.2 Generalising the Similarity Metrics).

7.3.3 Scale

In the research it is assumed that all castings are the same size. Scale therefore is one such useful measure and can be treated as a further modification in the light of experience with the results of the test performance graphs. Although two shapes may be structurally identical, one may in real life be of a different physical weight and size; this is where the importance of 'scale' comes into operation. Such a metric could weed out undesirable shapes from a match, although the shapes may be identical (for example, a 2000-ton ship propeller and a lightweight dinghy propeller). To achieve this, shapes can have an attribute 'size', and this would be an integer constant.

7.3.4 Qualitative Attributes

The role of qualitative attributes on a shape is an important aspect, which could support the analysis of differentiating between certain classes of shape (such as weight, size, manufacturability). Further work could include attributes of the whole shape, perhaps a textual description, and attributes at a lower component level. Qualitative attributes would allow a general search, perhaps using natural language;

for example, “show me all racing car wheels,” “show me all castings where weight >= 200 tons”). This would allow the Methods Engineer to browse through a database of castings, using queries relating to both whole shape and to problems with individual elements; for example: “show me all sea glands with the thin plate problem.”

7.3.5 Integration with Physical Modelling Systems

There are many physical modelling systems used in casting, which give excellent solidification simulations. However, their inherent fault lies in not being able to give valuable re-design advice. A casting may be sound, but a cheaper way to make a similar, or identical casting may have been cast several years ago. If *ShapeCBR* could be integrated with such physical modelling systems (such as SOLSTAR), the retrieval system would not only give sound solidification advice, but valuable design analysis could also take place. This would valuably aid the casting industry.

7.3.6 Automatic Shape Composition from CAD Models

The shapes created in the trial domain were composed manually using a prototype drawing system called *CastAID* (see Paper I in Appendix A). Ideally, this drawing process should operate automatically using algorithms to generate shape characterisations from CAD models (such as AutoCAD) automatically. The Methods engineer would then select a number of views through the shape, and the system would cut these slices from the CAD model and decompose them into elementary shapes. A PhD project is currently in progress at Greenwich University concerning this problem.

7.4 Conclusions

This chapter has drawn the thesis together with a summary of research carried out, contributions and an agenda for future research. The conclusion drawn from this thesis is that it is possible to automate the retrieval processes that a Methods Engineer uses in recalling previous similar casting design cases for a given target casting design case by using a natural shape decomposition scheme derived from knowledge acquisition, and the tasks involved in achieving this aim have been described.

Bibliography

American Foundrymen's Society website: www.afsinc.org

American Society for Metals, *Casting Design Handbook*, 1962

AutoCAD Release 13, Autodesk Inc, 1994. Website www.autocad.com

AutoLisp. See AutoCAD Release 13, Autodesk Inc, 1994. AutoLISP Reference, in Customisation Guide. pp: 159-578. Website: www.autocad.com

BCIRA - Cast Metals Development Centre. Casting Design - back to basics Part 1, *Engineering Designer*, pp: 12-15, March/April, 1992a

BCIRA - Cast Metals Development Centre. Casting Design - back to basics Part 2, *Engineering Designer*, pp: 12-15, May/June, 1992b

British Metal Casting Association website: www.martex.co.uk/taf/0466.htm

Crusader software, SCRATA, Sheffield, UK

Feedercalc, FOSECO Ltd, Tamworth, UK

MAGMAsoft, Giessereitechnologie GmbH, Werner-Heisenburg-Str. 14, D-5110 Alsdorf, Germany

MAVIS, Alphacast Software Limited, 56 Javelin Close, Duston, Northampton NN5 6PL, England

MOD (Ministry of Defence), Ship Department Publication 18, Design and Manufacture of Nickel-Aluminium Bronze Sand Castings, June, 1979

NOVACAST, Soft Center, S-372 25 Ronneby, Sweden

SIMULOR, Aluminium Pechiney, Aluval, BP27 38340 Voreppe, France

Aamodt, A; Plaza, E. Case-Based Reasoning: Foundational issue, methodological variations and system approaches, *AICom*, vol. 7, no.1, 1994

Aha, D. The omnipresence of case-based reasoning in science and application, *Knowledge-Based Systems*, Vol. 11 (5-6), pp. 261-273, 1998

Aha, D. Case-Based Reasoning, *Invited paper AIRIES '94 Workshop*, 1994

Alberts, L.K; Wognum, P.M; Mars, N.J.I. Structuring design knowledge on the basis of generic components, *Artificial Intelligence in Design '92*, Kluwer Academic Publishers, pp: 639-656, 1992

Althoff, K, Auriol, E, Barletta, R, Manugo, R. A review of industrial case-based reasoning tools, AI Intelligence, *Springer-Verlag*, 1995

Ballard, D.H; Brown, C.M. *Computer Vision*, Prentice Hall, 1982

Beeley, P.R; Smart, R.F (eds.) *Investment Casting*, The Institute of Materials, 1995

Beeley, P.R. *Foundry Technology*, Butterworth & Co, 1972

Bilgic, T. and Fox M. S. (1996) "Constraint-based Retrieval of Engineering Design Cases: Context as constraints", in Gero J. and F. Sudweeks (eds) Artificial Intelligence in Design '96.

Bradley, J; Gupta, U. A classification framework for case-based reasoning systems, *The New Review of Applied Expert Systems*, pp: 35-84,1995

Bradley, F.J; Adams, T.M; Gadh, R; Mirle, A.K. On the development of a model-based knowledge system for casting design, *Modeling of casting, welding and Advanced Solidification Processes VI*, pp: 161-168, 1993

Bridge, D. G. Defining and combining symmetric and asymmetric similarity measures, Advances in Case-Based Reasoning, *Lecture Notes in Artificial Intelligence 1488*, Springer-Verlag, 1998

Bronsvoort, W.F.; Bidarra, R; Dohmen, M; van Holland, W; de Kraker, K.J. Feature modelling for concurrent engineering. In: *Proceedings of the International Symposium on Tools and Methods for Concurrent Engineering 1996, 29-31 May, Budapest, Hungary*, Horvath, I and Varadi K (eds.), Technical University of Budapest, pp: 46-55, 1996

Bronsvoort, W.F; Jansen, F.W. Feature modelling and conversion - key concepts to concurrent engineering, *Computers in Industry*, 21, pp: 61-86, 1993

Brown, M; filer, N. Beauty vs. the beast: the case against massively parallel retrieval, *Progress in Case-Based Reasoning, First UK Workshop*, Springer-Verlag, pp: 42-58, 1995

Burkhard, H; Kuhnel, R; Pirk, P. Case based diagnosis in a technical domain, *Artificial Intelligence and Information Controls Systems of Robots '94*, pp: 241-246, 1994

Campbell, J. *Private e-mail communication*, October, 2000

Campbell, J. *Castings*, Butterworth-Heinemann Ltd, 1991

Case, K; Gao, J. Feature technology: an overview, *Int.J.Computer Integrated Manufacturing*, vol.6, nos. 1&2, pp: 2-12, 1993

Cha, M; Gero, J.S. Shape pattern recognition using a computable pattern recognition, A*rtificial Intelligence in Design '98*, Kluwer Academic Publishers, pp: 169-187, 1998

Chung, J.C.H; Patel, D.R; Cook, R.L. Feature-based modeling for mechanical design, *Computer and Graphics*, volume 14, number 2, pp: 189-199, 1990

Clegg, A.J. Foundry CAD/CAM developments: an overview, *The British Foundryman*, pp: 396-400, October 1986

Clifford, M.J. Foundry 2000 - the UK casting industry in the year 2000, *Foseco Foundry Practice*, no.23, pp: 2-5, November, 1992

Corbett, C.F. Right first time - with SOLSTAR. *Foseco Foundry Practice*, pp: 2-7, Number 218, December, 1989

Cowell, D.F; Knight, B; Preddy, K. Case-based decision support for the design of casting procedures, *Knowledge Based Systems for Civil and Structural Engineering*, (ed. B. Topping), pp: 251-259, 1994

Cowell, D.F; Knight, B; Preddy, K. A decision support system for the design of casting procedures, *Artificial Intelligence in Engineering*, 1993

Creese, R.C; Waibogha, S. Casting reject elimination using expert systems, *AFS Transactions*, pp: 617-620, 1987

Cross, M. Development of novel computational techniques for the next generation of software tools for casting simulation, *Modelling of Casting, Welding and Solidification Processes*, volume IV, pp: 115-123, 1993

Damski, J.C; Gero, J.S. A logic-based framework for shape representation, *Computer-Aided Design*, vol.28, no.3, pp: 169-181, 1996

Darwish, S.M; El-Tamimi, A.M. The selection of the casting process using an expert system, *Computers in Industry* 30, pp: 77-86, 1996

Darwish, S.M.H. Computer-assisted riser design and placement for casting, *Journal of Manufacture*, pp: 60-63, October, 1995

Dolsak, B; Jezernik, A; Bratko, I. A knowledge base for finite element mesh design, *Artificial Intelligence in Engineering*, 9, pp: 19-27, 1994

Duffy, A. The "what" and "how" of learning in design, *IEEE Expert*, pp: 71-76, May/June, 1997

Estrin, L. A deeper look at casting solidification software, *Modern Casting*, July, pp: 20-23, 1994

A, Ferguson and Bridge, D.G. Generalised Prioritisation: A New Way of Combining Similarity Metrics, Derek Bridge, Ruth Byrne, Barry O'Sullivan, Steven Prestwich and Humphrey Sorensen (eds.), *Procs. of Tenth Irish Conference on Artificial Intelligence & Cognitive Science*, pp.137-142, 1999

Falcidieno, B; Giannini, F. Automatic recognition and representation of shape-based features in a geometric modelling system, *Computer Vision, Graphics and Image Processing* 48, pp: 93-123, 1989

Gebhardt, F. Survey on structure-based case retrieval*, The Knowledge Engineering Review*, vol. 12:1, pp: 41-58, 1997a

Gebhardt, F; VoB, Gräther, Schmidt-Belz. Reasoning with complex cases, *Kluwer Academic Publishers*, 1997

Hedberg, S. New knowledge tools, *Byte*, July, 1993

Hennessy, D; Hinkle, D. Applying case-based reasoning to autoclave loading, *IEEE Expert*, volume 7, number 5, pp: 21-26, October, 1992

Higginbotham, G. Towards the achievement of virtual prototyping in solidification processes, *Modeling of Casting, Welding and Solidification Processes*, volume VII, pp: 115-123, 1995

Hill, F.S. *Computer Graphics*, Macmillan, 1990

Hill, J.L; Berry, J.T. Geometric feature extraction for knowledge based design of rigging systems for light alloy castings, *Modeling of Casting, Welding and Advanced Solidification Processes V*, 1991

Holt, A; Benwell, G.L. Case-based reasoning and spatial analysis, *URISA Journal*, pp: 27-36, 1996

Hua, K; Faltings, B; Smith, I. CADRE: case-based geometric design, *Artificial Intelligence in Engineering 10*, pp: 171-183, 1996

Huang, Y.S; Webster, P.D; Dean, T.A. CADCast Design and Production software for castings, *AFS Transactions*, vol.103, paper 95-18, pp: 69-73, 1995a

Huang, Y.S; Webster, P.D; Dean, T.A. Software for the design of castings and rapid production of patterns, *Proceedings of the 31st International Matador Conference*, Manchester, UK, pp: 563-569, 1995b

Jackson, J. E. *A User's Guide to Principal Components*, John Wiley and Sons, 1991

Jared, G.E.M; Dodsworth, J R. Solid Modelling, in *Principles of Computer-Aided Design*, (eds.) Joe Rooney and Philip Steadman, Pitman/Open University, 1987

Jolly, M. Overview and appraisal of numerical analysis software for the simulation of casting processes. *BICTA Bulletin* 23, 1996

Kluska-Nawarecka, S. Decentralized decisions system for recognizing casting defects, *Journal of Materials Science and Technology*, vol.4, no.1, 1996

Knight, B; Cowell, D; Preddy, K. An object-oriented support tool for the design of casting procedures. *Engineering Applications of Artificial Intelligence*, volume 8, number 5, pp: 561-567, 1995

Kolodner, J. *Case-based Reasoning*, Morgan Kaufmann, 1993

Kolodner, J.L An introduction to case-based reasoning. *Artificial Intelligence Review*, 6, pp: 3-34, 1992

Kotschi, R.M; Plutshack, L.A. An easy and inexpensive technique to study the solidification of castings in three dimensions, *AFS Transactions*, vol.89, pp: 601-610, 1981

Lenz, M; Burkhard, H.D; Bartsch-Sporl, B; Wess, S. Case-Based Reasoning Technology -- From Foundations to Applications. *Lecture Notes in Artificial Intelligence 1400*. Springer-Verlag, 1998

Lees, B; Corchado, J M. integrated case-based neural network approach to problem solving, in: Frank Puppe (Eds.): *XPS-99: Knowledge-Based Systems - Survey and Future Directions,* Lecture Notes in Computer Science, Vol. 1570, Springer-Verlag, pp: 157-166, 1999

Luby, S.C; Dixon, J.R; Simmons, M.K. Designing with features: creating and using a features database for evaluation of maufacturibility of castings. *ASME Computer Review*, pp: 285-292, 1988

Maher, M.L. and Gomez de Silva Garza, A. The adaptation of structural system designs using genetic algorithms. *Proceedings of the International Conference on Information Technology in Civil and Structural Engineering Design -- Taking Stock and Future Directions*, Glasgow, Scotland. August, 1996

Marir, F; Watson, I. Can CBR imitate human intelligence and are such systems easy to design and maintain? A critique, *Progress in Case-Based Reasoning, first UK Workshop*, Springer-Verlag, pp: 107-117, 1995

Marir, F; Watson, I. Case-based reasoning: a categorized bibliography*, The Knowledge Engineering Review*, volume 9:4, pp: 355-381, 1994

Mileman, T. Automating decisions in designing Castings, *34th Foundry Days, Brno, Czech Republic*, 30th September to 1st October, 1997

Mott, S. Case-based reasoning: market, applications and fit with other technologies, E*xpert Systems with Applications*, vol.6, pp: 97-104, 1993

Nagasawa, S; Miyata, Y; Murayama, M; Sakuta, H. Support system for finite element analysis, *Advances in Engineering Software*, 27, pp: 179-189, 1996

Nanda, N.K; Smith, K.A; Voller, V.R; Haberle, K.F. An heuristic based practical tool for casting design, *Modelling of Casting, Welding and Advanced Solidification Processes VII*, 1995

Nanda, K.N; Smith, K.A; Haberle, K.F; Voller, V.R. A knowledge-based computer tool for casting process design, *Journal of Manufacture*, pp: 27-30, September 1994

Natarajan, R; Chu, C.N; Kashyap, R.L. An integrated environment for intelligent design of castings. *Expert System Applications in Materials Processing and Manufacturing*, (ed.) M.Y.Demeri, The Minerals, Metals and Materials Society, pp: 143-157, 1989

Nealon, J.L; Firth, P. A knowledge-based approach to process planning: a case study, *IFAC Information Control Problems in Manufacturing Technology*, Madrid, Spain, pp: 559-564, 1989

Niyama, E; Uchida, T; Morikawa, M; Saito, S. A method of shrinkage prediction and its application to steel casting practice. *AFS Cast Metals Research Journal*, pp: 52-63, September, 1982

O'Halloran, J. 3D CAD, *CADCAM*, pp: 45-48, January, 1995

Orogo, C.D; Callihan, H.D; Sigworth, G.K; Kuhn, H.A. A vision of computer-aided casting in the year 2000, *Modern Casting*, pp: 20-24, October, 1993

Phelps, T.A; Heine, R.W; Uicker, J.J. Analysis of internal unsoundness casting defects using artificial intelligence techniques, *AFS Transactions*, 89-109, pp: 507-512, 1989

Pinker, S. Words and Rules: The Ingredients of Language, *Weidenfeld & Nicolson*, London 1999

Piwonka, T.S. Gating and Feeding Investment Castings, Chapter 6 in *Investment Casting*, Beeley, P.R; Smart, R.F (eds.), The Institute of Materials, ISBN: 0-901716-66-9, 1995

Plastock, R. A; Kalley, G. Schaum's Outline of Theory and Problems of Computer Graphics, McGraw-Hill, 1986

Prasad, T.V; Kondic, V. Designing feeding systems for investment castings, *Modern Casting*, September, 1994

Preddy, K.L.; Knight B.; Cowell D.F; Mileman, T.J. CAST-AID: A complementary partner to numerical modelling of casting simulation. Institute of British Foundrymen, *CASTCON '97 Proceedings*, Birmingham, England, 1997

Price, C.J; Pegler, I.S. Deciding parameter values with case-based reasoning, *Progress in Case-Based Reasoning*, (ed. Ian Watson), Springer-Verlag, pp: 121-133, 1995

Price, C.J; Peglar, I.S; Ratcliffe, M.B; McManus, A. From troubleshooting to process design: closing the manufacturing loop, *Lecture notes in Artificial Intelligence*, 1266, case-based reasoning research and development, Springer-Verlag, 1997

Price, C.J; Pegler, I.S; Bell, F. Case based reasoning in the melting pot, *Int. Journal of Applied Expert Systems*, vol. 1, no.2, 1993

Ransing, R.S; Srinivasan, M.N; Lewis, R.W., ICADA: Intelligent computer aided defect analysis for castings, *Journal of Intelligent Manufacturing*, 6, pp: 29-40, 1995

Raphael, B; Kumar, B. Indexing and retrieval of case in a case-based design system, *Artificial Intelligence for Engineering Design*, pp: 47-63, 1995

Ravi, B. Computer-aided casting design – past, present and future, *Technical paper for publication in the special issue of Indian Foundry Journal, Dept. of Mechanical Engineering, Indian Institute of Technology*, September, January 1999

Ravi, B. *Knowledge-based casting design*, official exchange paper for presentation at 62nd world foundry congress, Dept. Mechanical Engineering, Indian Institute of Technology, Bombay, India, 1996a

Ravi, B; Creese, R.C; Ramesh, D. Design for casting – a new paradigm for preventing potential problems, *Technical paper for publication in the AFS Transactions,* Dept. of Mechanical Engineering, Indian Institute of Technology, September, 1998

Ravi, B; Creese, R.C. Development of advanced software technology for metal casting, Project proposal report, *Department of Mechanical Engineering, Indian Institute of Technology, Bombay, India*, July 1996

Ravi, B; Srinivasan, M.N. Casting solidification analysis by modulus vector method, *International Cast Metals Journal*, vol.9, no.1, 1996

Ravi, B; Srinivasan, M.N. Hot spots in castings: computer-aided and experimental validation. *AFS Transactions*, 98, pp: 353-357, 1990

Ravi, B; Srinivasan, M.N. Hot spot detection and modulus computation by computer graphics, *56th World Foundry Congress paper*, pp: 12.3-12.8, Dusseldorf, May 1989

Rigout, C; Meyer, H; Charbonnier, A; Bourg, A. Round robin benchmark exercise for mould filling and solidification, *Modeling of Casting, Welding and Solidification Processes*, volume VII, 1995

Rowe, C, A. Blind risers: why they don't always work, parts 1 to 3, *Modern Casting*, February, March, April 1991

Ruddle, R.W. Risering - past, present and future, *AFS transactions*, pp: 269-280, 1971

Scherer, A; Berkel, T; Schlageter, G; Schultheiss, R. Using neural networks in a CAD/CAM application, pp: 253-262, *Techniques and Applications of Neural Networks*, (eds. P.J.G Lisboa, M.J. Taylor) Ellis Horwood, 1993

Sillen, R. Using Artificial Intelligence in the foundry, *Modern Casting*, pp: 34-37, December, 1991

Simpson, B.L. *Development of the Metal Castings Industry*, American Foundrymen's association, 1969

Sirilertworakul, N; Webster, P.D; Dean, T.A., A software package for the design and production of castings, *The Foundryman*, October, 1993a

Sirilertworakul, N; Webster, P.D; Dean, T.A; computer prediction of location of heat centres in castings, *Materials Science and Technology*, vol.9, pp: 923-928, October, 1993b

Sirilertworkakul, N; Webster, P.D; Dean, T.A. A knowledge base for alloy and process selection for casting, *Int. J. Mach. Tools Manufact.*, vol. 33, no.3, pp: 401-416, 1993c

Stoiljkovic, V.R; Mitrovic, A; Stoimenov, L; Milovanovic. EXCAST - Expert system for designing casting technology, *Conference paper, Moving Towards Expert Systems Globally in the 21st century*, published by cognizant communication corporation, Elmsford, NY, USA, p. 635-640, 1994

Sylvia, J.G. *Cast Metals Technology*, Addison-Wesley, 1972

Taylor, S.J. *An Investigation in Automation of Fire Field Modelling Techniques,* PhD thesis, University of Greenwich, 1997

Tu, J.S; Olinger, D.M; Hines, A.M. Computer-aided development of an investment casting process, *Journal of Manufacture*, pp: 29-32, October, 1993

Upadhya, G; Paul, A.J; Hill, J.L. Optimal design of gating and risering for castings: an integrated approach using empirical heuristics and geometric analysis, *Modeling of Casting, Welding and Advanced Solidification Processes VI*, pp: 135-141, 1993

van Holland, W; Bronsvoort, W.F. Assembly features and sequence planning, In: Product modeling for computer integrated design and manufacturing - P*roceedings Fifth IFIP WG5.2 Workshop on Geometric Modeling in Computer-Aided Design*, 18-23 May 1996, Airlie, USA, Pratt, M, Siram, R.D and Wozny, M.J (eds.), Chapman & Hall, London, pp: 275-284, 1996

Wang, Y; Ishii, N. A method of similarity metrics for structured representations, *Expert Systems with Applications*, Vol.12, No.1, pp: 89-100, 1997

Watson, I. An introduction to case-based reasoning. *Progress in Case-Based Reasoning*, First United Kingdom Workshop, Salford, January 12, Proceedings, Springer-Verlag pp: 3-16, 1995

Watson, I. *Knowledge Based Engineering*, Module 7 Workbook, Salford University, England, 1996

Webster, C.A.G. The use of images in a language- and culture-independent expert system for diagnosing pressure die-casting defects, *Journal of Materials Processing Technology*, 55, pp: 296-302, 1995

Webster, C.A.G; Weller, M; Sfantsikopoulos, M.M; Tsoukala, V.D. ALEXSYS - a prototype knowledge based expert system for the quality assurance of high pressure die castings, Database and expert systems applications, 4th international conference, *Dexa '93 Proceedings*, pp: 396-400, 1993

Wittgenstein, L. Philosophical investigations, *Blackwell*, 1953

Woodward, J.A.J; Corbett, J, An expert system to assist the design for manufacture of die cast components, *Engineering Design*, pp: 14-17, May/June, 1990

Wlodawer, R. *Directional Solidification of Steel Castings*, Pergamon Press, 1966

Wright, B.J. The role of the methods engineer, *The Foundryman*, pp: 377-378, December, 1992

Yeh, I. Case-based approaches for preliminary design of steel building frames, *Microcomputers in Civil Engineering*, volume 12, pp: 327-337, 1997

You, I.C; Chu, C.N; Kashyap, R.L. Expert system for castability evaluation: using a fixed-features based design approach, *Robotics and Computer-Integrated Manufacturing*, vol.6, no.3; pp: 181-189, 1989

Zhang, H.G; Webster, P.D; Dean, T.A. Computer aided design of feeders for casting, *Proceedings of the Institute of Mechanical Engineers*, pp: 279-287, vol. 208, 1994

Appendix A

Published Paper I

T J Mileman, B Knight, D F Cowell, K L Preddy. Cast-Aid: a decision support system for designing castings. Published in the *Proceedings of the 8th International Conference on Modeling of casting and welding processes*, San Diego, June 7-12, pp: 1087-1094, 1998

CAST-AID: A DECISION SUPPORT SYSTEM FOR DESIGNING CASTINGS

T J Mileman, B Knight, D F Cowell, K L Preddy*

Foundry Process Technology Group, The University of Greenwich, London SE18 6PF, U.K.

IRC Materials*,University of Birmingham, Birmingham, U.K.

Abstract

This paper describes a computer-based tool that models the expertise of a Methods Engineer, and allows for a higher degree of interaction and early feedback. The tool is intended to assist the engineer with decisions relating to the design of the casting. Of prime importance in the formation of these decisions is the construction of the so-called modulus model of the shape to be cast, which gives a crude approximation of the dynamic cooling and freezing order of the parts of the shape. Several computer-based models as support tools for the Methods Engineer have been attempted previously, usually based on the superposition of a regular grid of cells on the shape, and iterative calculation of freezing, cell by cell. In this paper, an alternative approach is described, which more closely models the Methods Engineer's expertise. There are two main elements to the expertise. First, a casting is systematically broken down into elements with known casting properties, and the suitability for casting based on the modulus model is applied to this assembly of elements. Second, is extensive knowledge of past cases (both sound and otherwise) of castings The support tool described here is naturally object-oriented, based on traditional classifications of shape elements, which have evolved over many years in casting design. This classification also provides natural keys for a case-based reasoning system. Experience with a prototype system known as CAST-AID is described, and the paper will demonstrate the progress to-date and its application to the Casting Industry.

Introduction

The successful casting of a predesigned shape is dependent on the skill and experience of the Methods Engineer, who performs a task known as *methoding* the design, to determine a number of key elements before manufacture. This task is concerned with the design of the casting process - the way in which the mold is filled with molten metal, and the subsequent freezing of the metal. The Methods Engineer must consider various problems associated with this process, the most important consideration being the shrinkage, which occurs as the metal freezes. The objective of good design is to avoid these problems by the suitable choice of the variable parameters at the engineer's disposal: placement of feeders and chills, orientation, running and gating. Various computer-based support tools may assist the design process. The process is an iterative one, in which tentative designs are evaluated using these tools. Initially there will usually be a dialogue with the client, where possible re-design of the shape for a more efficient casting may be suggested. The Methods Engineer will then decide on the orientation of the mold during filling, and design the positions and sizes of feeders and chills. For this task he/she will rely mostly on experience with hand calculations of the modulus of elements of the shape. The modulus of an element is the ratio of its volume to cooling surface area, modified by empirical shape specific correction factors [1], which can give an approximation of the solidification time taken for the element to cool. The design can be evaluated against various simulation models, and changes are made until the simulations are satisfactory.

Several software tools may be used to assist the methoding process. For the initial stages of methoding these tools need to be fast and easy to use: simple models based on the cooling modulus principle, or fast empirical mold-filling models. Among these are CRUSADER [2], FEEDERCALC [3] and SOLSTAR [4], which support the preliminary design stages, and slower, more detailed models such as MAGMASOFT [5] and SIMULOR [6], which support the simulation stages. CRUSADER and FEEDERCALC give numerical support on such aspects as feeder sizes and feeder distances, but do not attempt to give experiential advice, for example, re-design advice or mold orientation. SOLSTAR is often used as a fast solidification model, which can check a given design, or give information on feeder positioning. More advanced numerical software, using computational fluid dynamics techniques is under development [7]. However, only a few organisations can afford the costs of the software, as well as the work-hours needed to input the complex geometries that typify today's design expectations and this is especially the case with most foundries being classed as small manufacturers.

There is active interest in the casting industry in developing expert systems for the design of methoding systems [8,9,10,11]. Natarajan *et al* [8] describe a system which uses a simple geometrical solidification model to make a preliminary assessment of castability for a limited class of shapes. This is based on the cooling modulus model, but it works on a rectangular grid of cells, rather than on whole features – rather like SOLSTAR.

Other AI research has focused on geometrical feature extraction before such assessments [12,13,14]. Luby *et al*. [12] approaches this problem by defining a shape grammar, allowing the creation of designs with a vocabulary of familiar geometric features. The design is then evaluated for manufacturability by the construction of the modulus model from the features. Feature description is becoming recognised as a strong candidate for a single data representation for design, design analysis and manufacturing planning in the general context of total computer integrated manufacturing; see the survey article by Case and Gao [15].

The central idea of the work described in this paper is that the expertise of the Methods Engineer is based on a special decomposition of the shape to be cast into elements. Whereas other approaches to design support tools, e.g. [8] have allowed arbitrary decomposition, followed by a numerical estimation of the element moduli, the Methods Engineer uses familiar elements with well-defined moduli. So, for user acceptance alone, there are advantages in the construction of a modulus model that emulates the expert process. However, there are other advantages to be gained from taking a feature modelling approach, considering that much experiential knowledge of castability has grown up around such decompositions. A casting shape can be represented in a form of a relation graph that can be stored in a database and such a representation allows 'pattern recognition' techniques to be used on the database to provide design advice. Therefore the system has the architecture of a case-based reasoning (CBR) system.

This paper describes the principles on which this decomposition into elements is based, and its realisation as a prototype object-oriented support system called CAST-AID at the University of Greenwich. The uses of the system as a design support system, as a modulus tool, a knowledge-based advice system and a case-based reasoning tool are described.

2. The CAST-AID Advice Tool

Methods Engineers amass vast experience of most aspects of past and current manufacturing technology during their working life. How could this vast pool of knowledge and expertise be captured as not only to assist practising Methods Engineers in the foreseeable future, but also to enable many a lifetime's work to be readily available to future generations? We propose "CAST-AID", a system using the power of Information Technology currently available to incorporate some of the experiential knowledge of skilled casting technologists to assist Methods Engineers in developing castable component designs. The system is designed to fit into the process at an early stage, during the consultation between the customer and the foundry. The tool combines the following three main functions as a Methods Engineer Support Tool: -

- Using a PC-Based CAD package (AutoCAD) provide a Modulus Tool showing graphically a traditional decomposition of a shape with the freezing order of the elements (the modulus gradient)
- As a Knowledge Based Advice System, including suggestions on re-design for castability, common to the expertise gained by casting technologists during the 20^{th} century
- As a Case-Based Reasoning System, which, as the Methods Engineer builds up the required casting design from basic elements of known modulus, would retrieve previously stored cases appropriate to the design in hand.

It should be remembered that in practice the modules would be interacting with each other at any one time to reduce the preparatory work that the Methods Engineer has to do. The principles behind each of these three modules are briefly outlined as follows:

2.1 The Modulus Tool

The construction of a modulus model is one of the first decisions a Methods Engineer will make. Such a model will give useful information about the qualitative assessment of castability, is calculated rapidly and is helpful in avoiding gross shrinkage porosity. However, there are disadvantages [16], for instance the model cannot be used to predict microshrinkage.

This expert decision making process has been replicated as a front-end Graphical User Interface using a PC-based CAD package [17] and a high-level programming language AutoLISP [18]. Selecting components and gluing them together makes a new shape. The modulus tool shows a coloured display of the components, with a range of shades from red to blue representing a modulus gradient. From the display the Methods Engineer is able to identify the direction of solidification, find hot spots (areas of delayed freezing); and to decide on molding direction. In this way the Methods Engineer can get rapid feedback on potential problem areas.

2.2 The Knowledge-Based Advice System

Methods Engineers accumulate vast knowledge and experience of casting manufacture throughout their working lives. Would it not be desirable to capture some of this knowledge on a computer? Although there have been many software tools (such as FEEDERCALC and SOLSTAR) and numerous nomograms, usually the experience gained has been lost for future generations. Ideally such a system should offer re-design advice and point out possible casting problems, suggesting the best way to make changes. Also a system should know whether a similar design has been cast before (perhaps twenty years ago) and showing the best locations for feeders and chills and the orientation in the mold.

The Modulus Tool already described goes quite far as a design/decision support tool. But what it can’t do is to tell the Methods Engineer about previous designs and problems with the current design. How then could it be made to do this?

To achieve this we have extended the Modulus Tool to act as an advice system that contains information about previous designs and use that information to provide design advice for the current design. By taking a feature based approach the shape will be represented as a relation graph allowing ‘pattern recognition’ techniques to be applied on the graph. Design advice is triggered by patterns in the graph.

2.3 The Case-Based Reasoning System

Most successful applications of artificial intelligence in manufacturing have, to date, taken the form of expert systems encoding heuristic knowledge as sets of domain-specific “if-then” rules. If a particular problem domain can be modelled with such rules, this is the ideal solution. Unfortunately, few domains yield to such brute-force tactics and this has led to a growing trend towards using a complementary approach, namely case-based reasoning (CBR) [19]. People often solve problems by remembering similar problems from experience, and then adapting these successful solutions by suitable modifications to meet current needs. Case-based reasoning mimics this human decision process and is used to provide the advice CAST-AID gives. A ‘case’ in CAST-AID is considered as a previous casting with information about the outcome of the casting; for example, how difficult it was to cast, the problems experienced, how it was modified. All foundries have extensive documentation of previous cases, so for practical purposes these cases will have to be transferred to a case base. A library of cases will therefore contain an entire casting history.

3. An Illustrated Example

The prototype version of CAST-AID can handle rotationally symmetric castings plus a limited class of three-dimensional shapes. In both modes the engineer builds shapes from an extensible library of basic components. This library includes filleted L, T and cross-junctions, bars, plates and wedges.

The shape is constructed by repeated gluing of components. As each component is added, it influences the cooling surface area of adjacent shapes, so that the system needs to recalculate their moduli and this is displayed using a simple colour coding scheme. This, with a knowledge base of feeding distances and acceptable modulus gradients gives instant warning of potential shrinkage problems. The engineer can experiment by modifying components, or by adding or removing chills to alleviate such problems.

Let us now consider an example of CAST-AID in action. Figure 1 shows a Sea-Gland to be cast in Aluminium-Bronze using a CO_2-Silicate mold. As it is rotationally symmetrical, the two-dimensional mode is used. The engineer specifies a vertical axis of rotation and builds up the shape as shown in Figure 2 using five components. The system displays the empirical modulus values alongside.

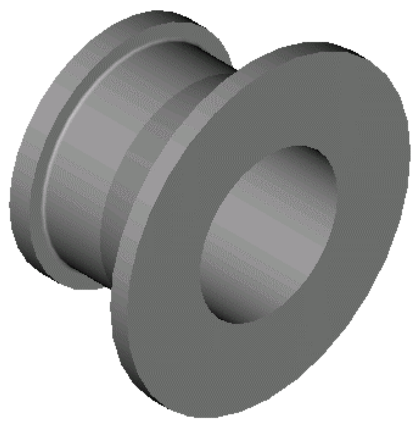


Figure 1 - Aluminium-Bronze Sea-Gland casting

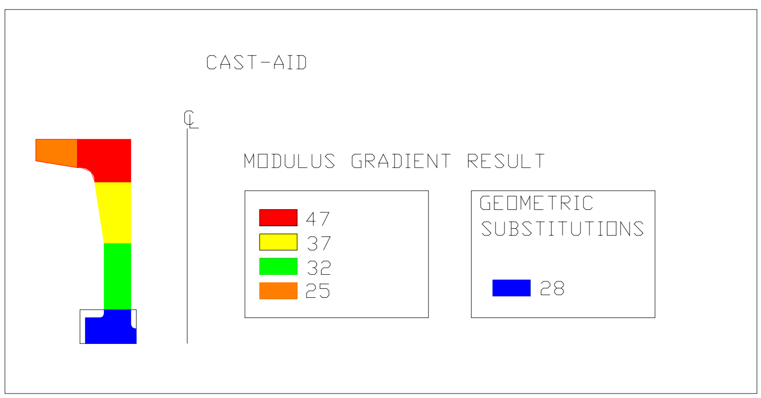


Figure 2 - CAST-AID build-up of Sea-Gland casting

On request the knowledge base checks for potential problems with this set-up. Two are identified:

1. The modulus of the foot (28mm) is too close to the modulus of the adjacent cylindrical section to promote directional solidification. Proposed solution: Reduce the foot modulus by adding chills.
2. The cylindrical section is too long to be fed adequately from the top feeders (which the system assumes by default). Proposed solution: Add blind feeders to the foot.

Although the current prototype is unable to provide advice about the "best" solution, the proposed CBR extension [20-22] would be used to retrieve similar related cases which would help the Methods Engineer pick the best alternative. Figure 4 shows part of two examples that might be retrieved by a search for similar cases to our sea-gland. Such examples would typically present a variety of "multimedia" information - photographs of castings, original blueprints and methoding calculations - bearing in mind that foundries typically keep records going back several decades - expert commentaries on why a particular casting was flawed, CAST-AID decompositions and designs, and results of numerical simulations obtained from packages such as SOLSTAR and MAGMASOFT.

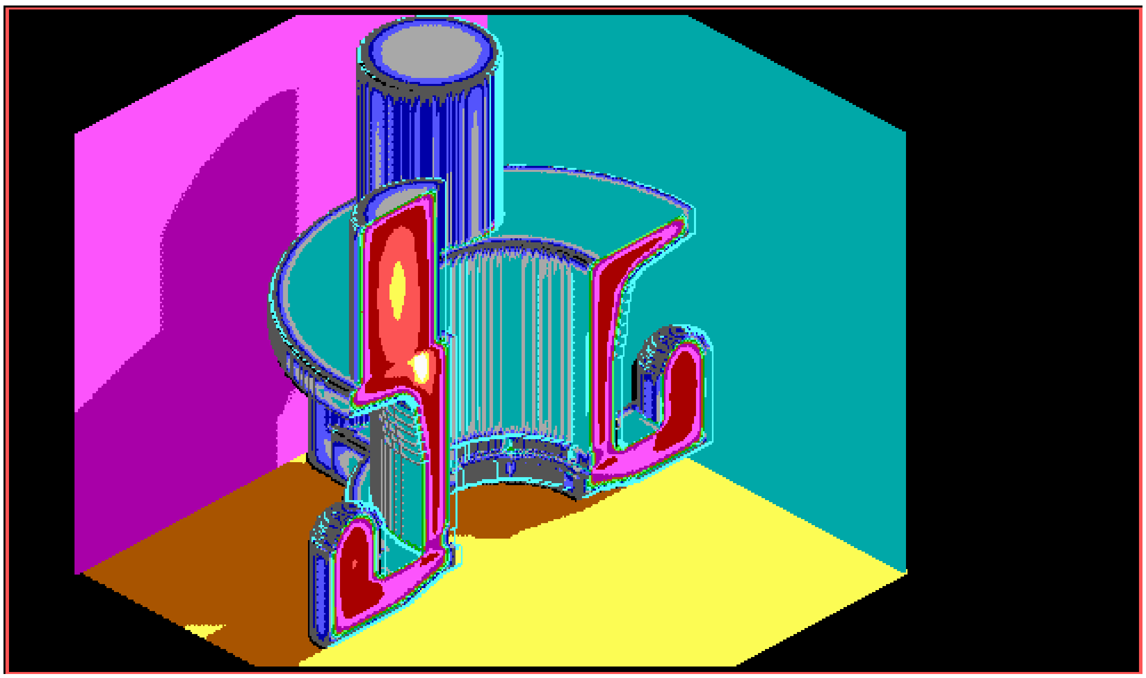


Figure 3 - SOLSTAR Class 1 Thermals. Compare with CAST-AID prediction in figure 2

The solution arrived at with the help of our prototype was checked by using SOLSTAR (Figure 3). This shows minimal shrinkage problems, verifying that the method is sound.

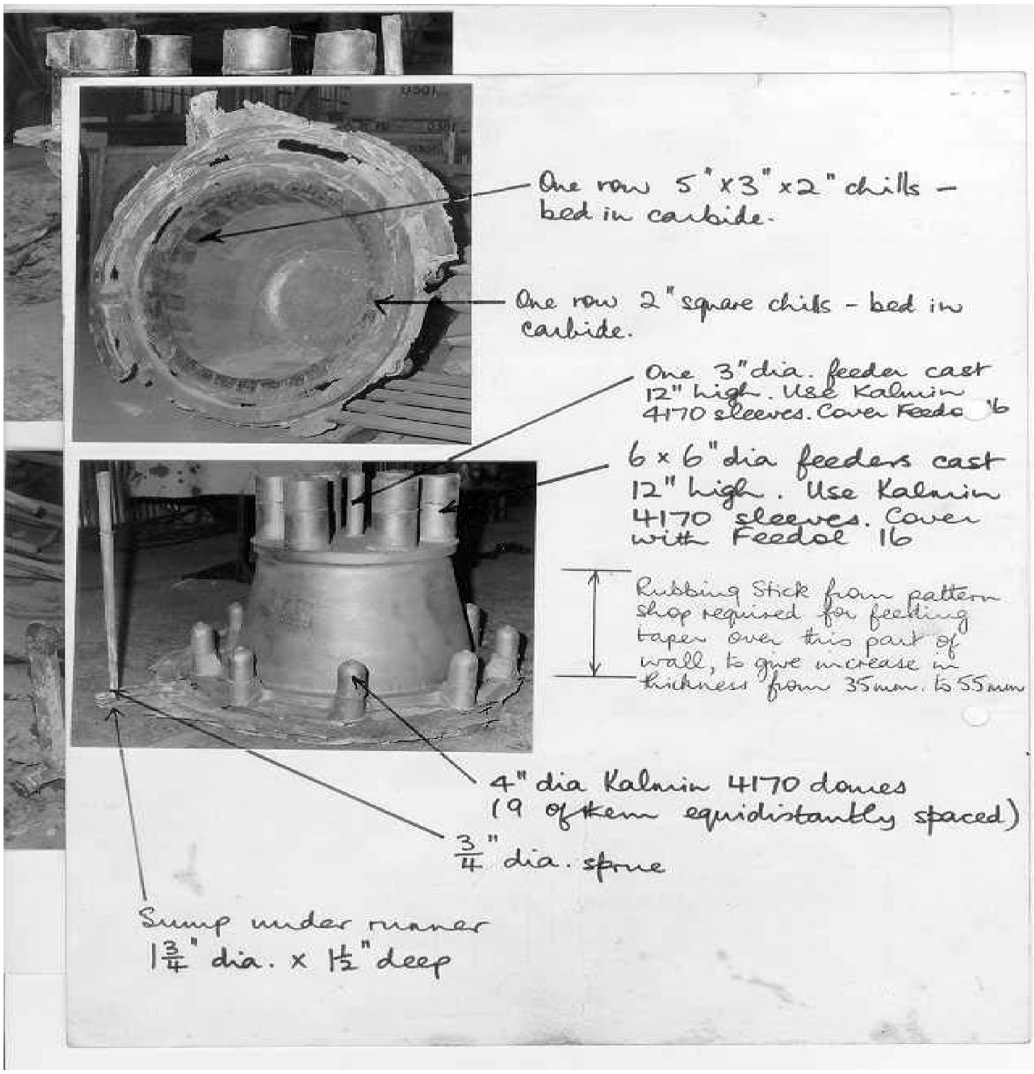


Figure 4 - Case-based reasoning retrieval example

4. Conclusion

In this paper we have described work in progress on a computer-based system to provide advice to Methods Engineers. The system employs three advice patterns: graphic display of component moduli, rule-based advice, and case-based reasoning. A prototype system is currently under evaluation, which deals with a limited set of component types: cross-junctions, L-junctions, plates, bars, cylinders, and rotationally symmetric analogues of these. As well as its use as a methoding tool, considerable interest has been shown in the system as a database retrieval engine for existing computerised records.

References

[1] Wlodawer, R. Directional Solidification of Steel Castings. Pergamon, Oxford, 1967
[2] CRUSADER software. SCRATA, Shefffield, U.K.
[3] FEEDERCALC. FOSECO Ltd, Tamworth, U.K
[4] SOLSTAR. FOSECO Ltd, Tamworth, U.K.
[5] MAGMA Giessereitechnologie GmbH, Werner-Heisenberg-Str. 14, D-5110 Alsdorf, Germany
[6] SIMULOR. Aluminium Pechiney, Aluval, BP27 38340 Voreppe, France
[7] Cross, M. "Development of novel computational techniques for the next generation of software tools for casting simulation." In Modelling of Casting, Welding and Advanced Solidification Processes IV ,pp.115-123. TMS, 1993
[8] Natarajan R., Chu C.M and Kashyap R.L. "An integrated environment for intelligent design of castings". In Expert Systems Applications in Materials Processing and Manufacturing (Edited by Demeri M.Y. The Minerals, Metals and Materials Society, 1989
[9] Sillen, R. "Using Artificial Intelligence in the Foundry." Modern Casting, December,1991
[10] Sirilertworakul, N., Webster P.D and Dean T.A. "A Knowledge Base for Alloy and Process Selection for Casting." Int. J. Mach Tools Manufact. 33, pp. 401-416. 1993
[11] Firth P. and Nealon R. "Process Planning: An Expert Systems Perspective." In Expert Systems – Theory & Applications, IASTED Int. Conf., Geneva, Switzerland, pp.170-174. 1987
[12] Luby S C., Dixon J.R and Simmons M.M. "Designing with features: creating and using a features database for evaluation of manufacturability for casting." ASME Comp. Rev., 285-292. 1988
[13] Woodward J.A.J. and Corbett, J. "An expert system to assist the design for manufacture of die cast components." Engineering Designer, 1990
[14] Chung J.C.H., Patel D.R., Cook R.K. and Simmons M.K. "Feature based modelling for mechanical design." Computer Graphics 14, pp. 189-199. 1990
[15] Case K, and Gao J. "Feature technology: an overview." Int. J. Computer Integrated Manufacturing 6, pp. 2-12. 1993
[16] Piwonka T.S. "Gating and feeding investment castings." In Investment Casting. Edited by Beeley P.R and Smart R.F; pp: 150-182. 1995
[17] AutoCAD Release 13, Autodesk Inc, 1994
[18] AutoCAD Release 13, Autodesk Inc, 1994. AutoLISP Reference, in Customisation Guide. pp: 159-578
[19] Barletta R. "An introduction to case-based reasoning." AI Expert, pp: 42-49. August 1991.
[20] Cowell D., Knight B., Preddy K. "A decision support system for the design of casting procedures," in Applications of Artificial Intelligence," in Eng. VIII, Vol. 1, Design Methods and Techniques (ed. Rzevski G *et al*) Computational Mechanics Publications, Elsevier Applied Science, pp: 251-257, 1993
[21] Cowell D., Knight B., Preddy K., "Case-Based Decision Support for the design for casting procedures", Knowledge Based Systems for Civil and Structural Engineering (ed. Topping B), pp: 251-259. 1994
[22] Cowell D., Knight B., Preddy K. "An object-oriented support tool for the design of casting procedures," in Engng. Applic. Artif. Intell. Vol.8, No.5, pp:561-557, 1995

Appendix B

Published Paper II

Tony Mileman, Brian Knight, Miltos Petridis, Keith Preddy, Patrick Mejasson. Maintenance of a Case-Base for the Retrieval of Rotationally Symmetric shapes for the Design of Metal Castings. *Advances in Case-Based Reasoning: 5th European Workshop, EWCBR 2000 Trento, Italy, September 6-9, 2000: Proceedings (Lecture Notes in Computer science)*, Blanzieri, E; Portinale, L (eds.), Springer-Verlag, 2000

Maintenance of a Case-Base for the Retrieval of Rotationally Symmetric shapes for the Design of Metal Castings

Tony Mileman, Brian Knight, Miltos Petridis, Keith Preddy, Patrick Mejasson

University of Greenwich, School of Computing and Mathematics, Park Row, Greenwich,
London SE10 9LS UK

{b.knight, m.petridis} @greenwich.ac.uk

Abstract

In this paper, we discuss the problem of maintenance of a CBR system for retrieval of rotationally symmetric shapes. The special feature of this system is that similarity is derived primarily from graph matching algorithms. The special problem of such a system is that it does not operate on search indices that may be derived from single cases and then used for visualisation and principle component analyses. Rather, the system is built on a similarity metric defined directly over pairs of cases. The problems of efficiency, consistency, redundancy, completeness and correctness are discussed for such a system. Performance measures for the CBR system are given, and the results for trials of the system are presented. The competence of the current case-base is discussed, with reference to a representation of cases as points in an n-dimensional feature space, and a Gramian visualisation. A refinement of the case base is performed as a result of the competence analysis and the performance of the case-base before and after refinement is compared.

1 Introduction

This paper addresses the problem of retrieval of rotationally symmetric shapes from a case base designed as a design assistant to be used in the metal casting industry. In a previous paper [7] a support tool was described, based on a traditional componentisation of rotationally symmetric shapes, with components of known cooling modulus. This paper describes further research into the use of this traditional componentisation as a shape representation for retrieval. The work concentrates on the connectivity of graphical components. There are of course other attributes which a practical case-based system can exploit, such as scale of cast objects and components, textual description, and traditional classifications of objects to be cast. There are plans to include these search indices to enhance the performance of the system described here.

Out of a range of solids processing methods for the mass production of components; for example, casting, forging and machining, casting is the generally the cheapest. However, the problem with casting is one of quality, which depends on the existence of casting design knowledge. The advantages of a CBR system, capable of containing detailed information on the design process for products, devolve from its ability to realise casting know-how as a valuable asset. The knowledge of how to cast a product soundly within tight cost constraints is the result of a huge investment on the part of industries, universities and government over many years. Although the value of design knowledge is widely recognised throughout the industry, the management of design knowledge is often ad hoc in some respects. Design histories are often lost, or banished to paper files that are difficult to search. Also, design engineers retire [27], or move away leaving inadequate design records.

There are many problems faced by a casting design engineer, centering on the physical freezing processes. Foremost among these is shrinkage in the mould, which can give rise to porosity and areas of structural weakness [1]. Other practical problems arise during pattern making and subsequent machining of the cast part. Many software tools have been developed to assist the designer. In a recent survey, Jolly [2] found that the foundry industry is looking for software that can not only predict problems that occur during metal solidification (such as shrinkage porosity) but also, having predicted these problems, to propose intelligent solutions to problems found. Current commercial casting software can be classified into two broad areas: intelligent knowledge-based systems (IKBS), and numerical simulations based on physical process models.

The IKBS approach regards the casting field as based on an expert perception of important factors, such as positioning chills and feeders, and determining shape orientation. The similarity metrics we have developed here operate directly on the geometrical features of a shape. Other successful systems, such as Wayland [26] and Clavier [28] operate in similar domains where one would think that being able to assess shape similarity is important, yet these systems still manage to be successful without directly using shape.

Although both Wayland and Clavier use shape, the techniques they use are not directly applicable to our research, mainly because they attempt to solve very different problems. Wayland for instance, is concerned with determining the machine settings of parameter values for die-casting machines, and to do this, the area and weight of a shape are important. In our research, we are concerned with configuring the geometric parameters for casting and this is more heavily dependent on casting geometry. The area and wall thickness, as used in Wayland, doesn't address these deep geometrical features. Similarly, Clavier, one of the early CBR systems is used for curing aerospace parts, performs exact textual matching to past cases, and again is not concerned with geometry.

Numerical modelling software packages are usually based on a finite-element code. The software can predict shrinkage formation in castings to reveal hotspots (such as MAVIS and DIANA) [3], and also to simulate filling and solidification (SIMULAR

[4], SOLSTAR [5]). Most of the newer packages allow a casting to be constructed in a CAD system (such as AutoCAD [6]) and swiftly imported, to reveal potential solidification problems. The main practical problem with these systems is that they only give feedback on a proposed design, and do not give advice on the design itself. The design of castings is a non-trivial problem. An engineer can easily design objects which are extremely similar in appearance, but one can be cast and the other one cannot. The ShapeCBR system currently provides general advice that the engineer can adapt in methoding a new case. Additional information can be gained about a casting by linking in solidification software, such as SOLSTAR, that can provide a visual mapping of the solidification processes to reveal problems such as hot spots and solidification shrinkage.

IKBS systems attempt to support an earlier stage in the design process. Numerous software tools such as those discussed in [7] have clearly demonstrated the usefulness of knowledge-based and other advanced heuristic-based programs for designing castings. Some of the commercial software packages available can calculate the position of feeders (NOVACAST [8]) and also to analyse geometric properties and to give suggestions to improve the design further (AutoCast [9]).

Although many prototype tools have demonstrated the efficacy of CBR in the domain of engineering and design [10-15], there is a scarcity of research for its use in the foundry industry. CBR can play an important role in intelligent casting software. Currently, there is one commercial CBR system [16] called Wayland, which we have briefly mentioned above, for the setting of parameters in pressure die-casting. This research has demonstrated that CBR has an exciting future in casting software.

The main problem for a CBR system is how to retrieve cases, where the retrieval must be based on shape. Although there are other possible search indices, for example the type of casting alloy, weight and general description of part (wheel, sea-gland, valve, engine bearing cap, etc.), these descriptions are too general for accurate retrieval. General classifications of shape components have been proposed; for example, Biederman's geons [17]. However, during this research, it became apparent during knowledge elicitation that a decomposition of shapes specific to the casting industry already existed in practice, [7, 18]. The research described here uses a graphical representation of shapes based on this decomposition as a foundation for shape retrieval.

In section 2 of this paper, the graphical representation is explained, and in section 3 similarity measures are proposed. Section 4 gives experimental results for the method based on a trial domain of rotationally symmetric objects. Section 5 discusses the competence of the trial case base, and shows how case base visualisation can assist in refining and maintaining the case base.

2 A graphical representation

In [7] a decomposition of a shape into a set of joined components was described. The decomposition is a natural one, used over many years by casting design engineers. It is based on a set of component types of significance in casting design. There are 8 main component types including Bar, L, T, X, Taper, Flange, Bespoke-Taper, and Bespoke-T. The componentisation process distinguishes two sets of component type: those that define the structure (L's, T's and X's) and those that join the first set together (bars and tapers). Using this classification, we may abstract a graphical representation of the structure of any shape S where the nodes are elements of the first set, and the arcs are elements of the second.

As an illustration, consider the rotationally symmetric shape shown in cross section in figure 1. A graph representation of this figure is given in Figure 2.

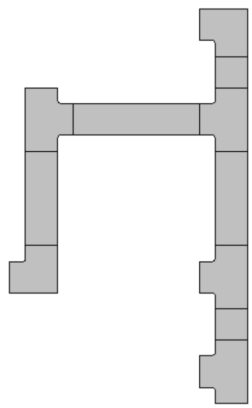


Figure 1. A casting, made of 3 component types (Bar, L, T)

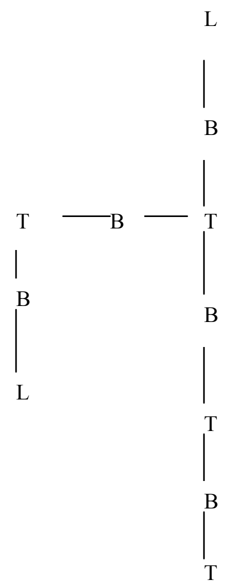


Figure 2. Representation of figure 1 as a graph

3 Similarity measures

Retrieval of shapes for casting design is an example of structure based case retrieval, as defined by Gebhardt [15]. For these systems, attributes representing complex structures are difficult to define, and similarity must be derived from structure directly. For the sub class of graphical structures, Gebhart reviews several retrieval systems. These include clique detection as in the Fabel component Topo [19], largest common subgraph [20] and hamming distance [21].

The research described here at Greenwich, we have used similarity measures based on features extracted from the structural graphs. Perfect similarity between shapes S1 and S2 is obtained when they have identical structural graphs. However for graphs that do not match completely, there are a number of features that can be extracted and compared. Each feature gives rise to a different similarity measure, representing a different case retrieval.

Correspondingly, there are a number of different problems associated with casting a shape, each connected with a different structural feature. Porosity tends to depend on specific local features, whereas machining problems tend to depend on global structure. The approach of this research has been to construct a retrieval tool to investigate the efficacy of the various metrics with respect to different casting design problems. The tool employs a generalised similarity measure $\sigma(S1,S2)$ between shapes S1 and S2, representing a weighted sum of the similarity measures based on different features extracted from the graphs of S1 and S2:

$$\sigma\,(S1,S2) = w_{comp}\sigma_{comp} + w_{mcs}\sigma_{mcs} + w_{cycle}\sigma_{cycle} + w_{leaf}\sigma_{leaf} \qquad \textbf{(1)}$$

Variation of the weights in this formula allows a general test of retrieval against any given casting problem.

The individual similarity metrics in (1) are defined as follows:

- σ_{comp}(S1,S2) is a measure based on the number of component types that are common to the two graphs. If two graphs are nearly identical, σ_{comp} will be close to 1. The length function is defined as length(S) = number of components in S, and the value of this metric is given by:

$$\sigma_{comp}(S1,S2) = \sum_{Comp} \frac{{S'_{comp}}^{2}}{\text{length(S1)length(S2)}} \qquad \textbf{(2)}$$

where S'_{comp} is the maximal number of common components of a particular type to graphs S1 and S2. Nevertheless, this metric does not take into account of the graph's topology. This is taken care of by the following metric:

- σ_{mcs} (S1,S2) is a measure based on the length of the maximum matching subgraph. If two graphs are nearly identical, σ_{mcs} will also be close to 1. This similarity metric is given by:

$$\sigma_{mcs}(S1,S2) = \frac{{S'}^{2}}{\text{length(S1)length(S2)}} \qquad \textbf{(3)}$$

where S' is the maximal common subgraph of S1 and S2, i.e. the largest graph which is a subgraph of both S1 and S2. The problem of finding S' is related to that of the well-known graph isomorphism problem. For small graphs of up to 10 arcs, a search based on direct comparison of all subgraphs of S1 with those of S2 is possible. For larger graphs a strategy based on a preliminary comparison of node types and degree can help to reduce the search time.

- σ_{ncycle} (S1,S2) is based on a count of elementary graph cycles:

$$\sigma_{cycles}(S1,S2) = 1 - \frac{|\text{ncycles(S1) - ncycles(S2)}|}{\max(\text{ncycles(S1)}, \text{ncycles(S2)})} \quad \textbf{(4)}$$

- σ_{nleaf} is based on a count of leaf nodes, and gives the number of branches to a tree:

$$\sigma_{leaves}(S1,S2) = 1 - \frac{|\text{nleaves(S1) - nleaves(S2)}|}{\max(\text{nleaves(S1)}, \text{nleaves(S2)})} \quad \textbf{(5)}$$

- $\sigma_{localstati}$ (i = 1,…n) are a set of statistics on local graphical features. For example, an important local feature might be two T's joined together. For this we may take localstatj to represent the number of T's joined by a joining component to another T. We define:

$$\sigma_{localstati} = |\ \text{stati(S1)} - \text{stati(S2)}\ | \quad \textbf{(6)}$$

3.1 The *ShapeCBR* System

The *ShapeCBR* software system has been developed at Greenwich to automate the process of matching a given target shape to a case in the Case Base. The case base is populated with cases containing information relevant to real metal casting experience. The information contained in each case relates to both a geometrical description of a real shape and domain specific information about the way that the shape was actually cast. Additionally, some cases may contain general expert advice relevant to casting the shape in a textual form. The system allows the user to retrieve a shape from the case base to match a target case, according to a match on the four contributing features as described in the previous section. Weighing factors can be applied by the user to attach varying importance to each of the similarity measures.

Figure 3 shows an example exercise of matching a target case to a retrieved case from the case base. Advice on positions of feeders and chills is annotated on the picture of the retrieved case.

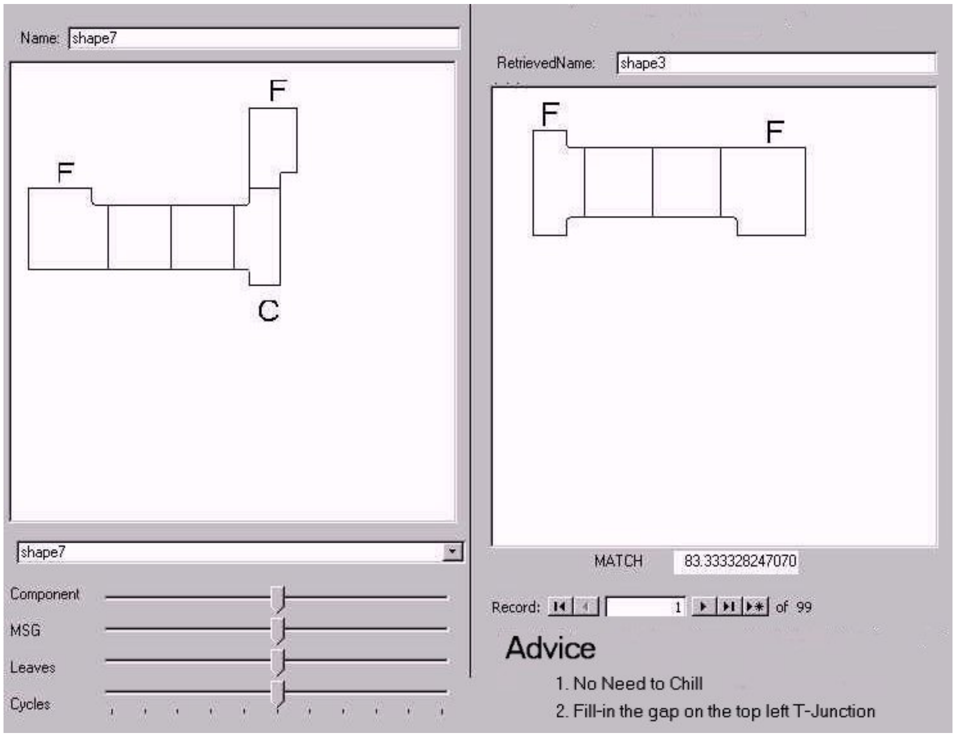


Figure 3. Matching a case to a target in the *ShapeCBR* system

Early feedback from the use of the system has been promising. The main problems for further development and enhancement of the case base in *ShapeCBR* are seen as:

Is the case base consistent and complete? When is it safe to delete cases? When should we add new cases? To help with these questions we used three tools: the system itself, a multidimensional scatter plot visualisation and a set of benchmark test cases.

The visualisations of the case base, and of the test set, are described in the next section. The distribution of the test set according to the whole case base gives a view of how representative the test cases were, and gives a level of confidence to the weightings used in the system. It may also be used to spot areas where the density of the system is high and where it may be possible to delete some cases.

The system itself can be used to validate whether a particular case is redundant with respect to any test set, by examining the performance measure when the case is removed. However, this is not a safe procedure for a fixed test set – for example, a case base consisting of the 20 test cases themselves would perform 100 % against themselves, but would have little capability for cases outside their scope.

4 Measuring performance

Testing the performance of knowledge based systems has been given much attention in the literature. Guida *et al* [22] in their paper on testing knowledge-based systems and performance measures, argue that performance measures are an essential tool for use in all stages of system development. They are particularly important during system maintenance, where knowledge is added, deleted and modified to effect system adaptation and improvement.

Following the ideas of Guida *et al*, we propose here several performance measures for the CBR system, which give a numerical estimate relative to any benchmark set of problems. In this section we describe the performance measures and give results for a trial case base consisting of 100 cases and tested against 20 new problems. In section 5 we show how the measures assist in the case base maintenance and refinement.

For the trial conducted here, we have taken the domain of shapes with rotational symmetry: wheels, armatures, cylinders, etc. This domain is coherent from a practical point of view, so that we can attempt to cover it with a limited case base. It is however sufficiently varied to encompass a wide range of casting problems.

Performance of the case base was assessed on several different measures. For a given target the retrieved set should provide the solution to (I) correct orientation, (II) the number and positions of feeders (III) the position of possible chills, (IV) the need for chills. (V) special problems encountered with this shape. For each of these problems, we can score how well the retrieved case presents the answer. For example, consider the target and retrieved shapes shown in Figure 4, which shows retrieved and test cases oriented, as they would be when pouring casting metal into the mould. The character F shows the position of the feeders, and C the chills. A1...A_n. shows presence of advice relating to the need for chills at that position.

In general we can see that there is an obvious visual match between the shapes. Without such a match none of the problems are solved, and we set all the measures to zero. However in this case the match is good enough to indicate the following performance measures:

(I) 100% - the orientation is correct. The convention in the figure is that the shapes were cast from the top as they are shown. Based on this, either shape will correctly predict the orientation of the other.
(II) 66% - only one feeder out of two. There is a matching pair of feeders, and one non-matching feeder.
(III) 100% - both chill positions are correct.
(IV) 100% - advice as to when chills are needed is correct.

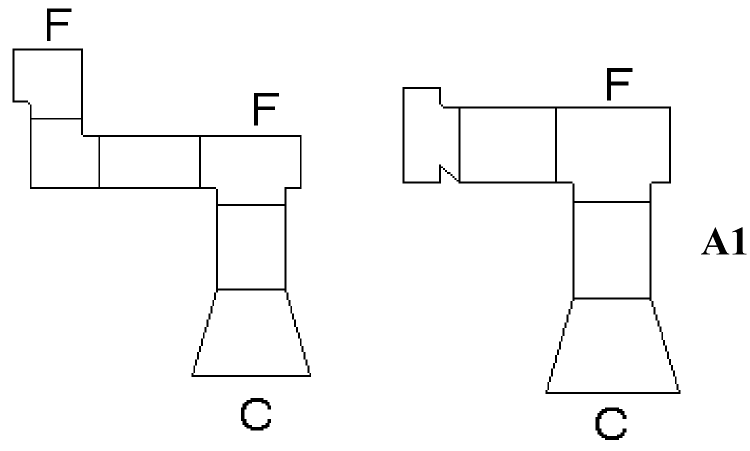


Figure 4. Target and Case showing orientation, feeders and chills. Advice A1: "chill this section if length > 4* breadth"

In situations where no obvious visual match may be made with the nearest case, we can widen the search to retrieve more cases, and leave the user to select the one with the best visual match. In such a mode of operation, the user is allowed to browse the nearest matches to look for the best advice.

4.1 Optimisation of Weights.

One use of the performance measure discussed in the first section is in the optimisation of the weights in equation (1). Initially, these were arbitrarily set equal; the idea being that the performance measure can be used to vary the weights to produce optimal performance. A frequently used technique in assessing performance is to measure how well a case base predicts itself. For a case base of 100, this requires us to enter 100 sets of performance data, corresponding to each case as predicted by its nearest match. As weights are changed then often the nearest match will change, and more validation data needs to be added. In reality it proved to be impractical to enter enough validation data to satisfy a wide range of weights required to allow an automatic search on weights for optimum performance.

It was decided to approach the problem incrementally, first selecting a subset of 20 cases, entering validation data for these, and performing an initial search on weights, to provide optimum values: w_{Comp0},w_{MCS0},w_{Cyc0},w_{Leaves}. Next, for weights near to these optimum values validation data was entered for the whole 100 cases, and the optimisation process repeated. Table 1 shows the performance on the whole case base (i) with equal weights, (ii) with the first set of optimised weights, and (iii) with final optimised weights.

The test set of 20 cases was selected independently and validated by an experienced casting engineer. Figure 5 shows a visualisation of the case base with equal weights, using the method of principal co-ordinates. The distribution of the test cases can also be seen on this plot, and seem to provide a fairly representative spread.

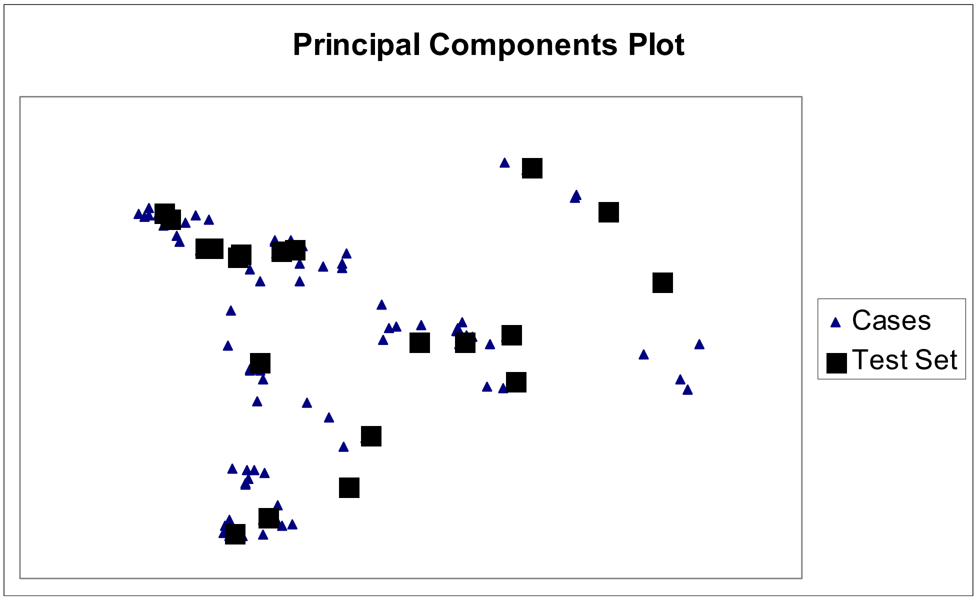


Figure 5. Plot showing distribution of Cases and test set.

	Equal weights	First optimisation	Final optimisation
Orientation	75.5%	81.5%	79.5%
Feeders	74.28%	76.8%	76.08%
chill positions	57.28%	56.27%	59.27%
Chill advice	82.5%	84%	84.50%

Table 1. Performance for whole case base, as weights are optimised

These performance figures can serve as a benchmark for future maintenance of the case base. We can require that future versions of the case base should perform at least as well on the original 100 cases as the performance figures in Table 1 show. This requirement can help to guide decisions on inclusion or deletion of cases.

5 The Competence of the System

An issue of importance in the assessment of CBR systems is that of system competence, that is the number of target problems that a given case base can solve [23, 24, 25]. The performance measures described in section 4 may be used to examine some questions connected with the competence of the case base. These questions are connected with the distribution of cases, i.e. with the problem-solving completeness, density and boundary of the case base. We summarise these questions as follows:

- Completeness: are there any gaps in the case base, i.e. problems that cannot be solved by retrieving near cases?

- Density: is the case base too dense in places? i.e. can we reduce the number of cases without affecting performance?

- Boundary: are there realistic castings representing cases outside our coverage group?

- When should we add new castings to the case base?

There are three tools that we can use to try to answer these questions. Firstly, there are the performance figures discussed above. These can serve as a benchmark for future maintenance of the case base. We can require that future versions of the case base should perform at least as well on the original 100 cases as the performance figures in Table 1 show. This requirement can help to guide decisions on inclusion or deletion of cases.

Secondly there are the principal component visualisations, as shown in figure 5, and three-dimensional scatter plots, which account for more of the distribution of cases. Figure 5 shows a definite boundary to the 100 cases. In fact, we can also see disjoint clusters, such as the shapes with 1 cycle (shown in figure 6). We can also see possible areas where the density of the case base could be high. However, since this is an (important) two-dimensional plot in a 100 dimensional space, care must taken to investigate these regions more carefully. The areas where there are gaps in the plot are difficult to interpret: are there missing realistic cases, or are they outside the boundary? However, the visualisation can give useful information about new cases. If a new case appears isolated in the middle of a gap, then it is probably a valuable addition to the system.

The third tool at our disposal is the retrieval system itself. As well as providing a tool to investigate further the cases indicated in the plot, it can also be used to find redundant cases. For example, we can run the system as in the performance test, retrieving on each of the cases in turn as targets. If 2 cases are always retrieved as a pair (i.e. one is 2nd nearest whenever the other is 1st nearest), and they both give the same performance measure on the target, then one of them may be deleted without affecting performance. Using this principle, we can safely eliminate some cases, and reduce density if needed.

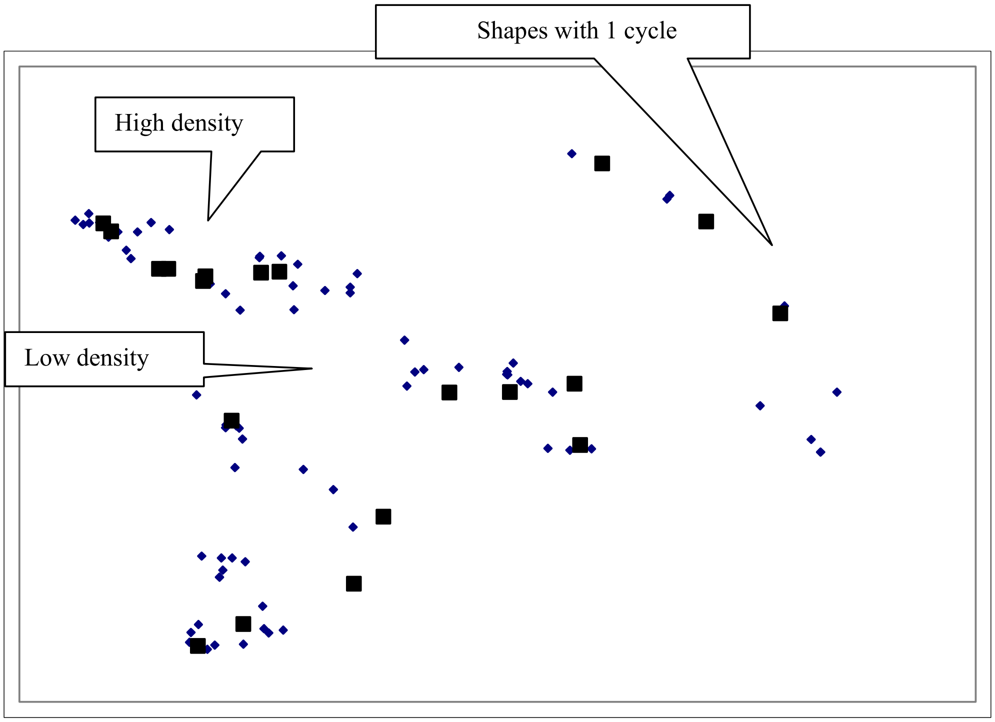


Figure 6. Clustering of the case base

6 Conclusion

In this paper we have described work on a case based system for the design of metal casting procedures. The key problem addressed by the work is the retrieval of rotationally symmetric shapes. The method proposed is based on a shape componentisation which is particular to the domain of casting problems. The shape componentisation gives rise to a graphical representation of shapes, from which similarity metrics may be abstracted.

The performance of the system has been measured for a sub-domain of rotationally symmetric shapes. A trial system consisting of 100 shapes has been constructed with the assistance of a casting design expert. The performance has been measured with respect to 3 key design decisions, and with respect to the retrieval of associated textual design advice. For the initial system, performance was assessed at between 59.27% and 84.5% of expert performance.

The paper also describes work done on the competence of the case base. A representation of the cases as points in an n-dimensional feature space is described, and a visualisation based on the first two principal components is presented. It is shown how refinements suggested by the visualisation may be made to affect case base density and coverage. The performance of the case base after refinement is given and compared to performance before refinement.

The results of these trials are encouraging, and indicate that the method is capable of extension into the full 3-D domain of shapes. Future work is planned to extend the trials to wider domains, including general 3-D systems. Work is also being planned for the integration of the system with physical modelling systems, such as SOLSTAR, to prototype the casting.

References

1. Campbell, J. "Castings", Buttterworth-Heinemann, 1991

2. Jolly, M. "Overview and appraisal of numerical analysis software for the simulation of casting processes", BICTA bulletin 3, 1996

3. Mavis, Diana. Alphacast software Ltd, England

4. Simular. Aluminium Pechiney, Aluval, BP27 38340 Voreppe, France

5. SOLSTAR. Corbett, C.F. "Right first time...with Solstar", Foundry Practice, Number 218, December, 1989

6. AutoCAD. Autodesk Inc. USA.

7. Knight, B; Cowell, D.; Preddy, K. "An object-oriented support tool for the design of casting procedures", In Engineering Applications of Artificial Intelligence, Volume.8, Number 5, pp: 561-567, 1995

8. NovaCast: Sillen, R. "Using artificial intelligence in the foundry", Modern Casting, December, 1991

9. Ravi, B. "Computer-Aided Casting Design - Past, Present and Future", Special Issue of Indian Foundry Journal, January, 1999

10. See p. 374, Marir, F; Watson, I. "Case-based reasoning: a categorized bibliography", The Knowledge Engineering Review, Vol.9: 4, pp: 355-381, 1994

11. Raphael, B; Kumar, B. "Indexing and retrieval of cases in a case-based design system", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 10, pp: 47-63, 1996

12. Hennessy, D; Hinkle, D. "Applying case-based reasoning to autoclave loading", IEEE Expert, pp: 21-26, October 1992

13. Kumar H.S; Krishnamoorthy, C.S. "Case-based reasoning in bridge design", pp: 197-205, Advances in Computational Structures Technology, Civil-Comp Press, Edinburgh, 1996

14. Yeh, I. "Case-Based approaches for preliminary design of steel building frames", Microcomputers in Civil Engineering, 12, pp: 327-337, 1997

15. Gebhardt, F. "Survey on structure-based case retrieval", The Knowledge Engineering Review, Vol.12: 1, pp: 41-58, 1997

16. Price, C.J; Peglar, I.S.; Ratcliffe, M.B; McManus, A. "From troubleshooting to process design: closing the manufacturing loop", Lecture Notes in Artificial Intelligence. 1266, Case-based reasoning research and development. Springer-Verlag, 1997

17. Biederman, I.; Hummel, J.E.; Cooper, E.E.; Gerhardstein, P.C. "From image edges to geons to viewpoint invariant object models: A neural net implementation", Applications of Artificial Intelligence X: Machine vision and robotics, 1992

18. Wlodawer, R "Directional solidification of steel castings", Pergamon, Oxford, 1967

19. Coulon, C.H. "General geometric and topological retrieval and adaptation (Topo)". In Borno, K, editor, Modulus for design support, Vol. 35 of Fabel report, pp: 33-445, GMD, Sankt Augustin, 1995

20. Tammer, E.C; Steinhofel, K; Schoner, S; Matuschek, D. "Anwendung des Konzepts der strutrellen Ahnlichkeit zum Fallvergleich mitterls Term- und Graph-representationen", Fabel report, 38, GMD, Sankt Augustin, 1995

21. Bunke, H; Messmer, B.T. "Similarity measures for structured representations". In Burkhard, H.D and Lenz, M, editors, Topics in Case-Based reasoning: First European workshop, EWCBR-93, selected papers; Lecture notes in Artificial intelligence, 837, pp: 106-118, Springer-Verlag, 1994

22. Guida, G; Mauri, G. "Evaluating performance and quality of knowledge-based systems: foundation and methodology", IEEE Transactions on Knowledge and Data Engineering, 5, 2, pp: 204-224, April 1993

23. Preece, A.D. "Towards a methodology for evaluating expert system", Expert Systems, 7(5), pp: 14-20, 1992

24. Racine, K; Yang, Q. "Maintaining unstructured case." Proceedings of the 2nd International Conference of Case-Based Reasoning, pp: 553-564, R1, USA, 1997

25. Smyth, B. "Case-Based Maintenance", Proceedings of the 11th International Conference on Industrial & Engineering Applications of AI & Expert Systems. Springer-Verlag, 1998

26. Price, C.J; Peglar, I.S "Deciding parameter values with case-based reasoning", Progress in case-based reasoning (ed. Ian D. Watson), Springer-Verlag, pp: 121-133, 1995

27. See page 123. Price, C.J; Peglar, I.S; Bell, F. Case based reasoning in the melting pot, Int. Journal of Applied Expert Systems, vol.1, no.2, pp.: 120-133, 1993

28. Hennessy, D; Hinkle, D. "Applying case-based reasoning to autoclave loading", IEEE Expert, pp: 21- 26, October 1992

Appendix C

Published Paper III

Tony Mileman, Brian Knight, Miltos Petridis, Don Cowell, John Ewer, Case-Based Retrieval of 3-D shapes for the design of metal castings.

NB: This is a version of the conference paper of Appendix B that has been submitted to the *Journal of Intelligent Manufacture*, Kluwer Academic Publishers. The paper has been accepted and is due to be published in the year 2001.

Case-Based Retrieval of 3-D shapes for the design of metal castings

Tony Mileman, Brian Knight, Miltos Petridis, Don Cowell, John Ewer
University of Greenwich, School of Computing and Mathematics, Park Row, London SE10 9LS UK

Abstract. This paper describes research into retrieval based on 3-D shapes for use in the metal casting industry. The purpose of the system is to advise a casting engineer on the design aspects of a new casting by reference to similar castings which have been prototyped and tested in the past. The key aspects of the system are the orientation of the shape within the mould, the positions of feeders and chills, and particular advice concerning special problems and solutions, and possible redesign. The main focus of this research is the effectiveness of similarity measures based on 3-D shapes. The approach adopted here is to construct similarity measures based on a graphical representation deriving from a shape decomposition used extensively by experienced casting design engineers. The paper explains the graphical representation and discusses similarity measures based on it. Performance measures for the CBR system are given, and the results for trials of the system are presented. The competence of the current case-base is discussed, with reference to a representation of cases as points in an n-dimensional feature space, and its principal components visualisation. A refinement of the case base is performed as a result of the competence analysis and the performance of the case-base before and after refinement is compared.

Keywords: Knowledge management, Case-Based Reasoning, Spatial reasoning, Casting design, Knowledge Based Systems, 3-D Shapes, CBR Competence, CBR Performance, Casting, Foundry.

Introduction

Out of a range of solids processing methods for the mass production of components, e.g. casting, forging, machining, etc. casting is the generally the cheapest. However, the problem with casting is one of quality, which depends upon the existence of casting design knowledge. The advantages of a uniform casting design system, capable of containing detailed information on the design process for products, devolve from its ability to realise casting know-how as a valuable asset. The knowledge of how to cast a product soundly within tight cost constraints is the result of a huge investment on the part of industries, universities and government over a long time period. Although the value of design knowledge is widely recognised throughout the industry, the management of design knowledge is often ad hoc in some respects. Design histories are often lost, or relegated to paper files which are difficult to search. Design engineers retire, or move away leaving inadequate design records. Information is not traded between companies, so that a new supplier must start from scratch without benefiting from previous designs, and foundries are not able to trade design knowledge which is no longer of any value to them.

This paper addresses the problem of retrieval of three dimensional shapes from a case base designed a design assistant to be used in the metal casting industry. In a previous paper [7] a support tool was described, based upon a traditional componentisation of 3D shapes , with components of known cooling modulus. This paper describes further research into the use of this traditional componentisation as a shape representation for retrieval.

Out of a range of solids processing methods for the mass production of components, e.g. casting, forging, machining, etc. casting is the generally the cheapest. However, the problem with casting is one of quality, which depends on the existence of casting design knowledge. The advantages of a CBR system, capable of containing detailed information on the design process for products, devolve from its ability to realise casting know-how as a valuable asset. The knowledge of how to cast a product soundly within tight cost constraints is the result of a huge investment on the part of industries, universities and government over many years. Although the value of design knowledge is widely recognised throughout the industry, the management of design knowledge is often ad hoc in some respects. Design histories are often lost, or banished to paper files that are difficult to search. Design engineers retire, or move away leaving inadequate design records.

There are many problems faced by a casting design engineer, centering on the physical freezing processes. Foremost among these is shrinkage in the mould, which can give rise to porosity and areas of structural weakness [1]. Other practical problems arise during pattern making and subsequent machining of the cast part. Many software tools have been developed to assist the designer. In a recent survey, Jolly [2] found that the foundry industry is looking for software that can not only predict problems that happen during metal solidification (such as shrinkage porosity) but also, having predicted these problems, to propose intelligent solutions to problems found. Current commercial casting software can be classified into two broad areas: numerical simulations based on physical process models, and intelligent knowledge based systems.

Numerical modelling software packages are usually based on a finite-element code. The software can predict shrinkage formation in castings to reveal hotspots (such as MAVIS and DIANA) [3], and also to simulate filling and soldification (SIMULAR [4], SOLSTAR [5]). Most of the newer packages allow a casting to be constructed in a CAD system (such as AutoCAD [6]) and swiftly imported, to reveal potential solidification problems. The main practical problem with these systems is that they only give feedback on a proposed design, and do not give advice on the design itself.

IKBS systems attempt to support an earlier stage in the design process. Numerous software tools as discussed in [7] have clearly demonstrated the usefulness of knowledge-based and other advanced heuristic-based programs for designing castings. Some of the commercial software packages available can calculate the position of feeders (NOVACAST) [8] and also to analyse geometric properties and to give suggestions to improve the design further (AutoCast) [9]

Although many prototype tools have demonstrated the efficacy of CBR in the domain of enginering and design [10-15], there is a scarcity of research for its use in the foundry industry. CBR can play an important role in intelligent casting software. Currently, there is one commercial CBR system [16] called CASPIAN, for the setting of parameters in pressure die-casting. This research has demonstrated that CBR has an exciting future in casting software.

The main problem for a CBR system is how to retrieve cases, where the retrieval must be based on shape. Although there are other possible search indices, for example the type of casting alloy, weight and general description of part (wheel, sea-gland, valve, engine bearing cap, etc.), these descriptions are too general for accurate retrieval. General classifications of shape components have been proposed, for example, Geons [17]. However, during this research, it became apparent during knowledge elicitation that a decomposition of shapes specific to the casting industry already existed in practice, [7, 18]. The research described here uses a graphical representation of shapes based on this decomposition as a foundation for shape retrieval.

In section 2 of this paper, the graphical representation is explained, and in section 3 similarity measures are proposed. Section 4 gives experimental results for the method based on a trial domain of rotationally symmetric objects. Section 5 discusses the competence of the trial case base, and shows how case base visualisation can assist in refining and maintaing the case base.

A graphical representation

In [7] a decomposition of a shape into a set of joined components was described. The decomposition is a natural one, used over many years by casting design engineers. It is based on a set of component types of significance in casting design. There are 8 main component types including Bar, L, T, X, Taper, Flange, Bespoke-Taper, and Bespoke-T. The componentisation process distinguishes two sets of component type: those that define the structure (L’s, T's and X’s) and those that join the first set together (bars and tapers). Using this classification, we may abstract a graphical representation of the structure of any shape S where the nodes are elements of the first set, and the arcs are elements of the second.

As an illustration, consider the rotationally symmetric shape shown in cross section in figure 1. A graph representation of this figure is given in figure 2.

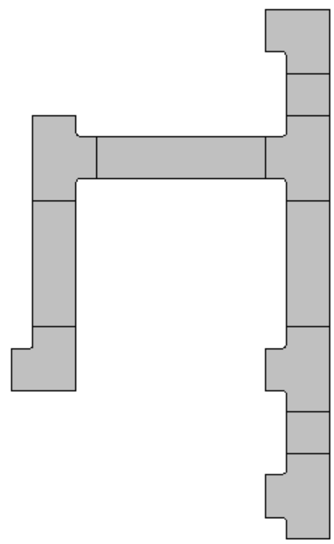


Figure 1 A casting, made of 3 component types (Bar, L, T)

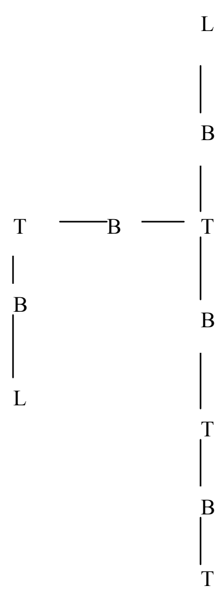


Figure 2 Representation of figure 1 as a graph

Similarity measures

Retrieval of shapes for casting design is an example of structure based case retrieval, as defined by Gebhardt [19]. For these systems, attributes representing complex structures are difficult to define, and similarity must be derived from structure directly. For the sub class of graphical structures, Gebhart reviews several retrieval systems. These include clique detection as in the Fabel component Topo [20], largest common subgraph [21] and hamming distance [22].

The research described here has used similarity measures based on features extracted from the structural graphs. Perfect similarity between shapes S1 and S2 is obtained when they have identical structural graphs. However for graphs that do not match completely, there are a number of features that can be extracted and compared. Each feature gives rise to a different similarity measure, representing a different case retrieval.

Correspondingly, there are a number of different problems associated with casting a shape, each connected with a different structural feature. Porosity tends to depend on specific local features, whereas machining problems tend to depend on global structure. The approach of this research has been to construct a retrieval tool with which to investigate the efficacy of the various metrics with respect to different casting design problems. The tool employs a generalised similarity measure σ(S1,S2) between shapes S1 and S2, representing a weighted sum of the similarity measures based on different features extracted from the graphs of S1 and S2:

$$\sigma\,(S1,S2) = w_{maxlen1}\sigma_{maxlen} + w_{ncycle}\sigma_{ncycle} + w_{nleaf}\sigma_{nleaf} + w_{localstat1}\sigma_{localstat1} + \ldots$$

Variation of the weights in this formula allows a general test of retrieval against any given casting problem.

The individual similarity metrics in 3.1 are defined as follows:

- σ_{maxlen} (S1,S2) is a measure based on the length of the maximum matching subgraph. If two graphs are nearly identical, σ_{maxlen} will be close to 1. The length function is defined as length(S) = number of components in S, and fm is given by:

  $$\sigma_{maxlen}(S1,S2) = \text{length}(S')/\max(\text{length}(S1),\text{length}(S2))$$

  where S' is the maximal common subgraph of S1 and S2, i.e. the largest graph which is a subgraph of both S1 and S2. The problem of finding S' is related to that of the well-known graph isomorphism problem (see e.g. [DolA 94]). For small graphs of up to 10 arcs, a search based on direct comparison of all subgraphs of S1with those of S2 is possible. For larger graphs a strategy based on a preliminary comparison of node types and degree can help to reduce the search time.

- σ_{ncycle} (S1,S2) is based on a count of elementary graph cycles:

$$\sigma_{ncycle} = |\ ncycle(S1) - ncycle(S2)\ |$$

- σ_{nleaf} is based on a count of leaf nodes, and gives the number of branches to a tree:

$$\sigma_{nleaf} = |\ nleaves(S1) - nleaves(S2)\ |$$

- $\sigma_{localstati}$ (i = 1,…n) are a set of statistics on local graphical features. For example, an important local feature might be two T's joined together. For this we may take localstatj to represent the number of T's joined by a joining component to another T. We define:

$$\sigma_{localstati} = |\ stati(S1) - stati(S2)\ |$$

Measuring performance

Testing the performance of knowledge based systems has been given much attention in the literature. Guida *et al* [23] argues that performance measures are an essential tool for use in all stages of system development. They are particularly important during system maintenance, where knowledge is added, deleted and modified to effect system adaptation and improvement.

Following the ideas of Guida *et al*, we propose here several performance measures for the CBR system, which give a numerical estimates relative to any benchmark set of problems. In this section we describe the performance measures and give results for a trial case base consisting of 100 cases, and used a test set of 20 cases against a case base of the other 80 cases. In section 5 we show how the measures assist in the case base maintenance and refinement.

For the trial conducted here, we have taken the domain of shapes with rotational symmetry: wheels, armatures, cylinders, etc. This domain is coherent from a practical point of view, so that we can attempt to cover it with a limited case base. It is however sufficiently varied to encompass a wide range of casting problems.

Performance of the case base was assessed on several different measures. For a given target the retrieved set should provide the solution to (I) correct orientation, (ii) the number and positions of feeders (III) the position of possible chills, (IV) the need for chills. (V) Special problems encountered with this shape. For each of these problems, we can score how well the retrieved case presents the answer. For example, consider the target shown in figure 1a, and the retrieved shape 1b. In general we can see that there is an obvious visual match between the shapes. Without such a match none of the problems are solved. However in this case the match is good enough to indicate the following performance measures:

(I) 100% - the orientation is correct
(II) 66% - only one feeder out of two

(III) 100% - chill positions are correct
(IV) 100% - advice as to when chills are needed is correct
(V) 50% - special advice in the retrieved case applies to the target.

In situations where no obvious visual match may be made with the nearest case, we can widen the search to retrieve more cases, and leave the user to select the one with the best visual match. Such a system is semi-automatic, but still practically very useful. In this situation we expect the performance indicators to improve, as is in fact shown in figure 3.

Tests were carried out, using a case base of 100 shapes. The tests were made on a set of 20 independently produced cases, which were validated by an experienced casting engineer. The results, showing average scores over the 20 targets are shown in table 1, both for completely automatic retrieval (nearest case), and for semi-automatic retrieval (best visual match on 5 nearest cases). Several trials have been conducted using different weights. The best results obtained so far are given in table 1.

Retrievals	Automatic	semi-automatic
Orientation	75%	97.5%
Feeders	69%	93%
Chill positions	60%	80%
Chills needed	80%	98%
Special advice	66%	75%

Table 1. Results of first trial

The Competence of the System

An issue of importance in the assessment of CBR systems is that of system competence, that is the number of target problems that a given case base can solve [24-26]. The performance measures described in section 4 may be used to examine some questions connected with the competence of the case base. These questions are connected with the distribution of cases, i.e. with the problem-solving completeness, density and boundary of the case base. We summarise these questions as follows:

- Completeness: are there any gaps in the case base, i.e. problems that cannot be solved by retrieving near cases?
- Density: is the case base too dense in places? i.e. can we reduce the number of cases and without affecting performance?
- Boundary: are there realistic castings representing cases outside our coverage group?

The approach we have taken here for considering the distribution properties of the case base is to represent cases as points in the multidimensional space of features given in section 3. All of these features give rise to meaningful dimensional integer constants, and the space is closely connected with the similarity measure employed in retrieval. Distribution can be assessed visually by examining 2-D plots of cases on pairs of features. However, this is difficult to assess in view of the large number of pair-wise plots necessary for the complete space. A better view is obtained by looking at the principal axes of the space, as shown in figure 3, which has the advantage of combining all features.

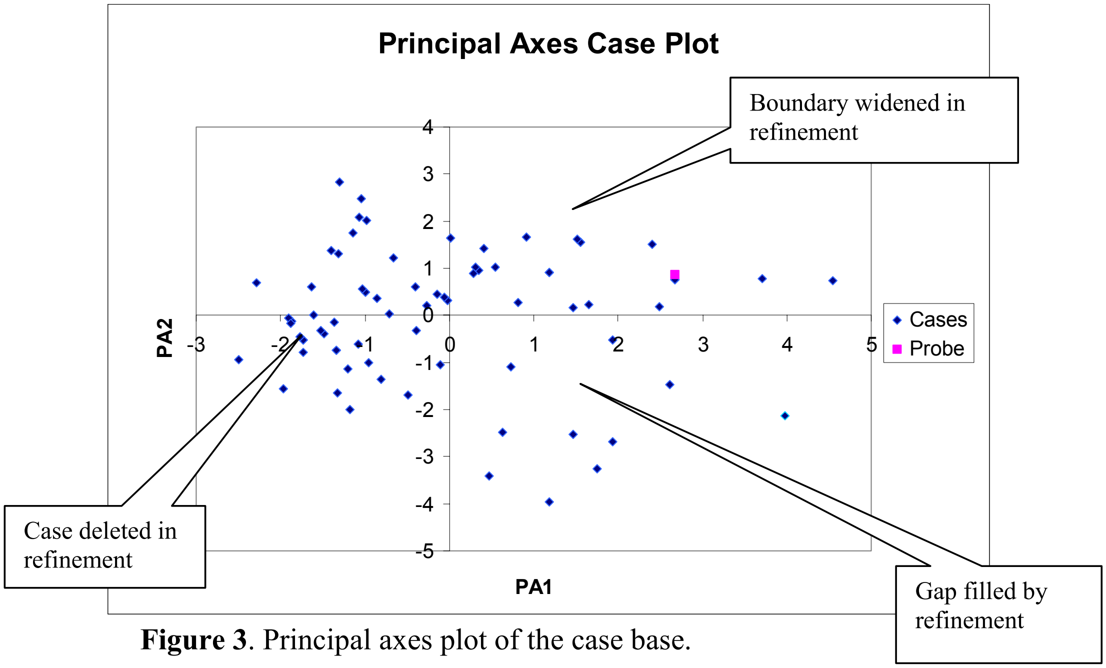


Figure 3. Principal axes plot of the case base.

Figure 4 shows the initial case base from which the results shown in table 2 were obtained. The plot gives the impression of a single coverage group of cases, contained within a convex boundary. Within the boundary there is evidence of gaps, and areas of high density. Three experiments were suggested by this visualisation.

First we reduced the number of cases in areas of high density. We were able to delete 12 cases, and obtained slightly reduced performance figures given in table 2:

retrievals	Automatic	semi-automatic
orientation	67.5%	92.5%
feeders	64.2%	93%
chill positions	60%	80%
chills needed	80%	98%
special advice	66%	75%

Table 2. Performance figure after deleting the 12 cases.

Secondly we found shapes to fill the gaps, 4 cases were added. Performance improved as in table 2. Finally we found realistic shapes representing points which widened the boundary. The principal axis visualisation of the case base after these three refinements is shown in figure 4.

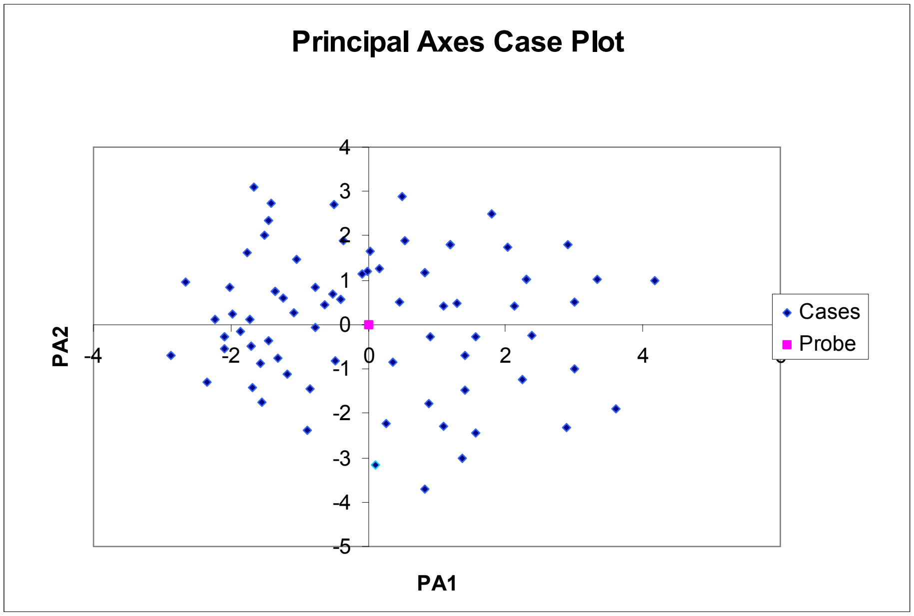


Figure 4. Principal Axis plot

Conclusion

In this paper we have described work on a case based system for the design of metal casting procedures. The key problem addressed by the work is the retrieval of 3D shapes. The method proposed is based on a shape componentisation which is particular to the domain of casting problems. The shape componentisation gives rise to a graphical representation of shapes, from which similarity metrics may be abstracted.

The performance of the system has been measured for a sub-domain of rotationally symmetric shapes. A trial system consisting of 100 shapes has been constructed with the assistance of a casting design expert. The performance has been measured with respect to 3 key design decisions, and with respect to the retrieval of associated textual design advice. For the initial system, two modes of system operation were tested: automatic and semi-automatic. Performance for the automatic mode was assessed at between 66% and 75% of expert performance. For the semi-automatic mode performance rose to between 75% and 97.5% of expert performance.

The paper also describes work done on the competence of the case base. A representation of the cases as points in an n-dimensional feature space is described, and a visualisation based on the first two principal components is presented. It is shown how refinements suggested by the visualisation may be made to affect case base density and coverage. The performance of the case base after refinement is given and compared to performance before refinement.

The results of these trials are encouraging, and indicate that the method is capable of extension into the full 3D domain of shapes. Future work is planned to extend the trials to wider domains, including general 3D systems. Work is also being planned for the integration of the system with physical modelling systems to prototype the casting.

References

[1] Campbell, J. "Castings", Buttterworth-Heinemann. 1991

[2] Jolly, M. "Overview and appraisal of numerical analysis software for the simulation of casting processes", BICTA bulletin 3, 1996

[3] Mavis, Diana. Alphacast software Ltd. England.

[4] Simular. Aluminium Pechiney, Aluval, BP27 38340 Voreppe, France

[5] Solstar. Corbett, C.F. "Right first time...with solstar", Foundry Practice, Number 218, December, 1989

[6] AutoCAD. Autodesk Inc. USA.

[7] Knight, B; Cowell, D.; Preddy, K. "An object-oriented support tool for the design of casting procedures", In Engineering Applications of Artificial Intelligence, Volume.8, Number 5, pp: 561-567, 1995

[8] NovaCast: Sillen, R. "Using artificial intelligence in the foundry", Modern Casting, December, 1991

[9] Ravi, B. "Computer-Aided Casting Design - Past, Present and Future", Special Issue of Indian Foundry Journal, January, 1999

[10] See p. 374, Marir, F; Watson, I. "Case-based reasoning: a categorized bibliography", The Knowledge Engineering Review, Vol.9: 4, pp: 355-381, 1994

[11] Raphael, B; Kumar, B. "Indexing and retrieval of cases in a case-based design system", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 10, pp: 47-63, 1996

[12] Hennessy, D; Hinkle, D. "Applying case-based reasoning to autoclave loading", IEEE Expert, pp: 21-26, October 1992

[13] Kumar H.S; Krishnamoorthy, C.S. "Case-based reasoning in bridge design", pp: 197-205, Advances in Computational Structures Technology, Civil-Comp Press, Edinburgh, 1996

[14] Yeh, I. "Case-Based approaches for preliminary design of steel building frames", Microcomputers in civil engineering, 12, pp: 327-337, 1997

[15] Gebhardt, F. "Survey on structure-based case retrieval", The Knowledge Engineering Review, Vol.12: 1, pp: 41-58, 1997

[16] Price, C.J; Peglar, I.S.; Ratcliffe, M.B; McManus, A. "From troubleshooting to process design: closing the manufacturing loop", Lecture Notes in Artificial Intelligence. 1266, Case-based reasoning research and development. Springer-Verlag. 1997

[17] Biederman, I.; Hummel, J.E.; Cooper, E.E.; Gerhardstein, P.C. "From image edges to geons to viewpoint invarient object models: A neural net implementation", Applications of Artificial Intelligence X: Machine vision and robotics, 1992

[18] Wlodawer, R "Directional solidification of steel castings", Pergamon, Oxford, 1967

[19] Gebhart, F. "Survey on structure-based case retrieval", The Knowledge Engineering Review, Vol.12: 1, pp: 41-58, 1997

[20] Coulon, C.H. "General geometric and topological retrieval and adaptation (Topo)". In Borno, K, editor, Modulus for design support, Vol. 35 of Fabel report, pp: 33-445, GMD, Sankt Augustin, 1995

[21] Tammer, E.C; Steinhofel, K; Schoner, S; Matuschek, D. "Anwendung des Konzepts der strutrellen Ahnlichkeit zum Fallvergleich mitterls Term- und Graph-representationen", Fabel report, 38, GMD, Sankt Augustin, 1995

[22] Bunke, H; Messmer, B.T. "Similarity measurere for structured representations". In Burkhard, H.D and Lenz, M, editors, Topics in Case-Based reasonin: First European workshop, EWCBR-93, selected papers; Lecture notes in Artificial intelligence, 837, pp: 106-118, Springer-Verlag, 1994

[23] Guida, G; Mauri, G. "Evaluating performance and quality of knowledge-based systems: foundation and methodology", IEEE Transactions on knowledge and data engineering, 5, 2, pp: 204-224, April 1993

[24] Preece, A.D. "Towards a methodology for evaluating expert system", Expert Systems, 7(5), pp: 14-20, 1992

[25] Racine, K; Yang, Q. "Maintaining unstructured case." Proceedings of the 2nd International Conference of Case-Based Reasoning, pp: 553-564, R1, USA, 1997

[26] Smyth, B. "Case-Based Maintenance", Proceedings of the 11th International Conference on Industrial & Engineering Applications of AI & Expert Systems. Springer-Verlag, 1998

[27] Dolan A Aldous , J " Networks and Algorithms: An Introductory Approach" John Wiley & Sons; ISBN: 0471939927, 1994

Appendix D

This appendix contains Visual Basic code for calculating the Maximum Common Subgraph metric of Chapter 5 - Section 5.1.2

The Visual Basic Code for the Maximum Common Subgraph Metric

```
Sub match(Sgraph As Cgraph, Tgraph As Cgraph, Snode As Integer, Tnode As
Integer, count As Integer, Slabel() As Integer, TLabel() As Integer)

Dim index1 As Integer
Dim index2 As Integer
Dim spath As Integer, tpath As Integer, no_pairs As Integer
Dim maxcount As Integer, imaxcount As Integer
Dim depth As Integer
Dim curNeighS As Integer, curneighT As Integer
Dim test As Boolean
Dim sArray(1 To 4) As Integer
Dim tArray(1 To 4) As Integer
Dim pairs As New pairing

Dim pathpair() As Integer

If Sgraph.Node(Snode).type <> Tgraph.Node(Tnode).type Or isIn(Snode, count,
Slabel()) Or isIn(Tnode, count, TLabel()) Then ' could get more complex
   count = 0
Else
   spath = 0
   For index1 = 1 To Sgraph.Node(Snode).noLinks
      If Not isIn(Sgraph.Node(Snode).Neigh(index1), count, Slabel()) Then
         spath = spath + 1
         sArray(spath) = Sgraph.Node(Snode).Neigh(index1)
      End If
   Next index1
   tpath = 0
   For index1 = 1 To Tgraph.Node(Tnode).noLinks
      If Not isIn(Tgraph.Node(Tnode).Neigh(index1), count, TLabel()) Then
         tpath = tpath + 1
         tArray(tpath) = Tgraph.Node(Tnode).Neigh(index1)
      End If
   Next index1
   Call addBlacklist(Snode, Slabel(), count + 1)
   Call addBlacklist(Tnode, TLabel(), count + 1)
   If spath > 0 And tpath > 0 Then
   Call pairs.init_couples(spath, tpath)

   Call findPerm(sArray(), tArray(), spath, tpath, 1, pairs)
   imaxcount = 0
   For index1 = 1 To pairs.get_counter
      maxcount = 0
```

```
        For index2 = 1 To pairs.get_couples
            depth = maxcount + count + 1
            Call pairs.getnextcouple(index1, index2, curNeighS, curneighT)


            Call match(Sgraph, Tgraph, curNeighS, curneighT, depth, Slabel(), TLabel())
            maxcount = maxcount + depth
        Next index2
            If maxcount > imaxcount Then imaxcount = maxcount
    Next index1
    End If
    count = imaxcount + 1


End If
End Sub
```

SubProcedure for Calculating the Maximum Common Subgraph

```
Sub matchMaxCSG(output() As Single)

Dim indexo1 As Integer, indexo2 As Integer
Dim index1 As Integer
Dim index2 As Integer
Dim count As Integer, maxcount As Integer
Dim Slabel(30) As Integer
Dim TLabel(30) As Integer



For indexo1 = 1 To no_targets
    For indexo2 = 1 To no_casebase
        count = 0
        maxcount = 0
        For index1 = 1 To targetcases(indexo1).no_nodes
            For index2 = 1 To casebase(indexo2).no_nodes
             ; call the algorithm
                Call match(targetcases(indexo1), casebase(indexo2), index1, index2, count,
Slabel(), TLabel())
                If count > maxcount Then maxcount = count
            Next index2
        Next index1
         ; now calculate the metric
```

```
        output(indexo1, indexo2) = 100 * maxcount * maxcount /
targetcases(indexo1).no_nodes / casebase(indexo2).no_nodes
        'PicDis.Print "The maximum common subgraph is " & CStr(maxcount)
        'PicDis.Print "The matching is " & CStr(100 * maxcount * maxcount /
myGraph1.no_nodes / myGraph2.no_nodes) & " %"
    Next indexo2
Next indexo1


End Sub
```

The Data Structures for Representing the Graph

```
Type Cnode
    type As Integer
    noLinks As Integer
    Neigh(1 To 4) As Integer ; maximum 4 neighbours (X)
End Type

Type Cgraph
    name As String
    no_nodes As Integer
    Node(1 To MAXNODES) As Cnode
    no_cycles As Integer
    no_leaves As Integer
End Type
```

Appendix E

This appendix contains AutoLisp code for calculating the Component metric of Chapter 5 - Section 5.1.1

The AutoLisp Code for the Component Matching Metric

```
; CBR component-metric
; compares shape1 (s1) with shape2 (s2) returning percentage
; of similarity of components.
(defun cbr1 (s1 s2)

   (* (* (is-in s1 s2) (is-in s2 s1)) 100)

) ; end cbr1



; returns a percentage of how much of shape1 is in shape2
(defun is-in (shp1 shp2 / cnt loop comp-type)

   (setq cnt 0.0)
   (setq loop 0)
   (repeat (length ALL-COMPONENTS)

      (setq comp-type (nth loop ALL-COMPONENTS)) ; got a component type

      (setq cnt (+ cnt (comp-match comp-type shp1 shp2)))

      (setq loop (1+ loop)) ; loop for next component type

   ) ; end repeat

     cnt ; return percentage

) ; end is-in


; fit function. Determines how
; s1=list of components in shape1
; s2=list of components in shape2
; example. S1=<TTTT>. S2=<TTT>. Value returned is 3, the fit number,
; since 3 T components are shared between both shapes
(defun fit (s1 s2 / l high low)

  (setq l (order s1 s2))

  (setq low  (nth 0 l))  ; get lowest value of a,b
  (setq high (nth 1 l))  ; get highest value of a,b
```

```
  (- high (- high low))  ; now return the fit number

) ; end fit



; given a component type in shape1,
; return the percentage of how much that component-type is in shape2
(defun comp-match (component-type s1 s2 / sum-s1 sum-s2)

    (setq sum-s1 (type-in-shape s1 component-type))  ; how many times the
component type is in s1
    (setq sum-s2 (type-in-shape s2 component-type))  ; how many times the
components type is in s2

    (/ (fit sum-s1 sum-s2) (length s1) ) ; return percentage

) ; end



; given a shape s and a component type c
; return n, where n  is the number of times
; component type c is in shape s
; end type-in-shape
(defun type-in-shape (s c / loop count comp)

  (setq loop 0)  ; loop for next component
  (setq count 0) ; stores a count of how many of type C is in S
  (repeat (length s)

    (setq comp (nth loop s))  ; get next component in the shape

    ; if the type matches c then increment count
    (if (= c (fetch-comp-type comp))
       (setq count (1+ count))
    ) ; end if

    (setq loop (1+ loop))

  ) ; end repeat

   (float count) ; return number of times c is in s

) ; end type-in-shape
```

Appendix F – Software Implementation

This appendix has further details about the programs involved in the CBR system showing how they operated to produce the case base and their execution for CBR retrieval.

System Implementation and Operation

The project is implemented as a system comprising of two main programs:

1. *CastAID*: An AutoLisp program that allows 2-D shapes that represent casting slices to be drawn and saved. This program was used first to create a case base of 100 methoded castings.

2. *ShapeCBR*: A MS Access program that handled the CBR process of shape matching, and used the files generated by *CastAID*.

The programs are examined briefly below:

CastAID

CastAID acts as a front-end Graphical User Interface using a PC-based CAD package (AutoCAD r13) and a high-level programming language *AutoLisp* that provides a Methods Engineer the means to create 2-D engineering drawings using the principle of shape decomposition. Initially the Methods Engineer will be presented with a blank screen and by selecting from a library of components (for example, cross-junctions, L-junctions, bars, and taper) a shape can be assembled. Each shape is saved as an *AutoCAD* drawing and a *CastAID* description of the shape – see Paper I in Appendix A for further details of the program's operation. *CastAID* stores a casting shape using the definitions shown on the next page.

AutoLisp Component Definition

This is the *Autolisp* definition of the component object that is used to store the components that make up a shape.

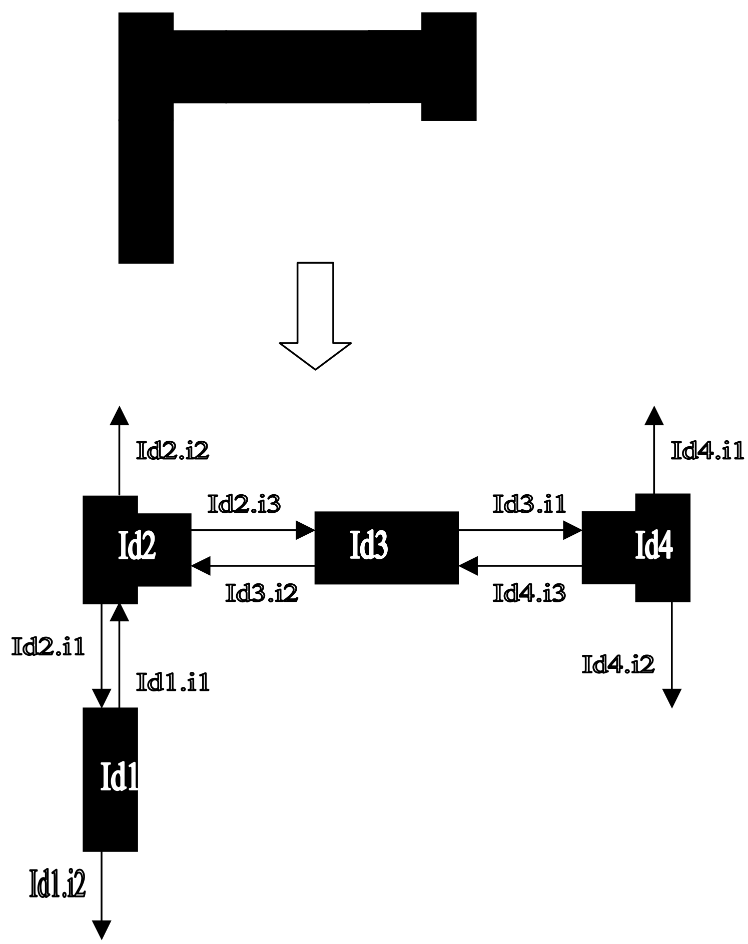
```
; define component table
        (setq COMPONENT-TABLE
            '("id" "type" "geometry" "origin"
        "x" "y" "z" "modulus"
        "numinterfaces" "chills" "feeders")
        )
```

AutoLisp Interface Definition

This is the *AutoLisp* definition of the interface table that stores the connections between components that make up a shape:

```
; set interface table
        (setq INTERFACE-TABLE
          '("component id" "interfacenum"
        "origin" "x" "y" "z" "connects-id"
        "connects-interface-num" "angle")
        )
```

As an example of this, the figure below shows part of the wheel of Figure 4.9 in Chapter 4, in terms of components and interfaces. Component identifiers are shown on the component as Id_n, and the interface linkages (i_n) are shown as connections between components. For example, Id1 (a bar) has two interfaces: i1 and i2. Interface 1 (i1) is not connected to any component, but interface 2 (i2) of Id1 (Id1.i2) connects to the T-junction (Id2); for clarity, the connecting interface number is not shown.



The next figure below shows the data model in terms of component and interface objects for this shape.

Component(Id1, Bar, *geometry*, *o,x,y,z*, *modulus*, numinterfaces=2, *chills*, *feeders*)
Interface i1. (Id1, 1, *o,x,y,z*, connects-to-id Id2, connects-to-interface i1, *angle*)
Interface i2. Not connected

Component(Id2, T-Junction, *geometry*, *o,x,y,z*, *modulus*, numinterfaces=3, *chills*, *feeders*)
Interface i1. (Id2, 1, o*,x,y,z*, connects-to-id Id1, connects-to-interface i1, *angle*)
Interface i2. Not connected.
Interface i3. (Id2, 3, *o,x,y,z*, connects-to-id Id3, connects-to-interface i2, *angle*)

Component(Id3, Bar, *geometry*, *o,x,y,z, modulus*, numinterfaces=2, *chills*, *feeders*)
Interface i1. (Id3, 1, *o,x,y,z*, connects-to-id Id4, connects-to-interface i3, *angle*)
Interface i2. (Id3, 2, *o,x,y,z*, connect-to-id Id2, connects-to-interface i3, *angle*)

Component(Id4, T-Junction, *geometry*, *o,x,y,z*, *modulus*, numinterfaces=3, *chills*, *feeders*)
Interface i1. Not connected.
Interface i2. Not connected.
Interface i3. (Id4, 3, o,x,y,z, connects-to-id Id3, connects-to-interface i3, *angle*)

AutoLisp Representation of the Sea Gland

This is a completed example of the sea-gland of Figure 2 in Paper I (Appendix A) in *CastAID* format:

Components:

("Sea-Gland" ("43" "**AXIS**" (0.0 2.0) (-5.0 16.0 0.0) (-4.0 16.0 0.0) (-5.0 15.0 0.0) (-5.0 16.0 -1.0) nil 3 nil nil) ("31" "**Taper**" (3.0 7.0 4.0) (6.0 26.0 -7.95994e-016) (6.0 25.0 -7.95994e-016) (7.0 26.0 -1.04092e-015) (6.0 26.0 1.0) nil 2 nil nil) ("2F" "**L-Junction**" (4.0 4.0 0.2) (1.0 26.0 4.28612e-016) (2.0 26.0 1.83691e-016) (1.0 25.0 4.28612e-016) (1.0 26.0 -1.0) nil 2 nil nil) ("2C" "**Taper**" (3.0 8.0 4.0) (1.0 21.0 1.83691e-016) (2.0 21.0 6.12303e-017) (1.0 20.0 1.83691e-016) (1.0 21.0 -1.0) nil 2 nil nil) ("25" "**Bar**" (3.0 8.0) (4.0 13.0 0.0) (4.0 12.0 0.0) (3.0 13.0 0.0) (4.0 13.0 -1.0) nil 2 nil nil) ("21" "**Flange**" (4.0 2.0 3.0 1.0 0.2 0.2) (0 0 0) (1 0 0) (0 1 0) (0 0 1) 1.0 3 nil nil))

Interfaces:

("Sea-Gland" ("43" 3 (2.0 -3.0 0) (2.0 -4.0 0) (2.0 -3.0 1) (1.0 –3.0 0) nil nil 0) ("43" 2 (5.0 6.0 0) (6.0 6.0 0) (5.0 6.0 1) (5.0 5.0 0) nil nil 0) ("43" 1 (5.0 15.0 0) (5.0 16.0 0) (5.0 15.0 1) (6.0 15.0 0) "21" 3 0) ("31" 2 (2.0 0 0) (3.0 0 0) (2.0 0 1) (2.0 -1 0) "2F" 1 0) ("31" 1 (1.5 7.0 0) (0.5 7.0 0) (1.5 7.0 1) (1.5 8.0 0) nil nil 0) ("2F" 2 (2.0 5.0 0) (1.0 5.0 0) (2.0 5.0 1) (2.0 6.0 0) "2C" 2 180) ("2F" 1 (5.0 2.0 0) (5.0 3.0 0) (5.0 2.0 1) (6.0 2.0 0) "31" 2 0) ("2C" 2 (2.0 0 0) (3.0 0 0) (2.0 0 1) (2.0 -1 0) "2F" 2 0) ("2C" 1 (1.5 8.0 0) (0.5 8.0 0) (1.5 8.0 1) (1.5 9.0 0) "25" 2 180) ("25" 2 (0 1.5 0) (0 0.5 0) (0 1.5 1) (-1 1.5 0) "2C" 1 0) ("25" 1 (8.0 1.5 0) (8.0 2.5 0) (8.0 1.5 1) (9.0 1.5 0) "21" 2 0) ("21" 3 (0 1.0 0) (0 0.0 0) (0 1.0 1) (-1 1.0 0) "43" 1 0) ("21" 2 (2.5 5.0 0) (1.5 5.0 0) (2.5 5.0 1) (2.5 6.0 0) "25" 1 0) ("21" 1 (5.0 2.0 0) (5.0 3.0 0) (5.0 2.0 1) (6.0 2.0 0) nil nil 0))

Each CastAID shape was stored in the format: <shapename>.CAID. Also, associated with each *CastAID* description is an AutoCAD drawing of the casting, in <filename>.DWG format; a standard CAD representation used by AutoCAD.

The one hundred cases created by *CastAID* were placed into one file called SHAPES.TXT for later access by *ShapeCBR*. Following this, the program GRAPH.LSP was executed to generate a file of graphs of the shapes, as below:

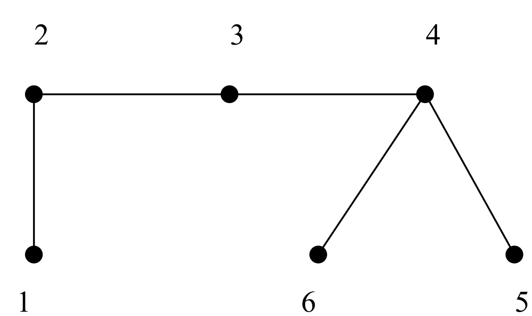
Graph Generation

An *AutoLisp* program called GRAPH.LSP read SHAPES.TXT and produced a file called GRAPH.TXT file, containing a graphical description of each of the 100 shapes. The GRAPH.TXT file was used in the Visual Basic graph program (see Appendix D for the code). Each graphical representation of a shape is stored using the following structure:

shape *a*:
Number of nodes,
Type, number of links *n*
Linked node 1
Link node *n*

Where type is an arbitrary number thus: L-Junction=0, T-Junction=1, X-Junction=2, Taper=3, Bespoke-Taper=4, Bar=5, Flange=6, Bespoke T-Junction =7.

For example, consider the graph below:



Component Type at Node:

Node 1 is a Bar

Node 2 is an L-Junction

Node 3 is a Bar

Node 4 is a T-Junction

Node 5 is a Bar

Node 6 is a Bespoke-Taper

The annotated graph format produced by GRAPH.LSP for this single is shown below:

6	← Number of graph nodes in the following order:
5,1	← Bar, 1 connection. Node 1
2	*(connects to node 2)*
0,2	← L-junction, 2 connections. Node 2
1	*(connects to node 1)*
3	*(connects to node 3)*
5,2	←Bar, 2 connections. Node 3
2	*(connects to node 2)*
4	*(connects to node 4)*
1,3	← T-Junction, 3 connections. Node 4
3	*(connects to node 3)*
5	*(connects to node 5)*
6	*(connects to node 6)*
4,1	←Bespoke-Taper, 1 connection. Node 5
4	*(connects to node 4)*
5,1	←Bar, 1 connection. Node 6
4	*(connects to node 4)*

Metric Result File

To avoid costly re-calculations in the *ShapeCBR* program (see below) a pre-processing program calculated the: components, mcs, leaves and cycles metrics for the 100 shapes before the CBR program was run (see **ShapeCBR** below). The following programs were used in this pre-processing program:

> *CBRGraph:* a visual Basic Program for computing the maximum common subgraph similarity metric; see Chapter 6 – Section 5.1.2 for the pseudo-code and Appendix D for the Visual Basic code. This program uses the GRAPH.TXT file, as described earlier in Graph Generation in this Appendix.
>
> *ComponentCBR*: An AutoLisp program for computing component similarity. This program is detailed in Appendix E.

The purpose of generating the metric results file was to pre-calculate all the metrics, instead of 'on the fly' during the CBR retrieval process. The reason why this was necessary is shown when the similarity of two graphs was calculated, which took approximately one minute. Comparing the graphs of the whole case base of one hundred cases against a target case would therefore take: 100 cases*1 minute ≈ 1.6 hours; this is impracticable for test purposes.

The results were stored in a text file called MetricsResults.TXT, and later read into a MS Access table in *ShapeCBR*. To determine the similarity of, for example case 56 and case 78, it was necessary to search to the Metric Results table for these cases, read the metric results and multiply the optimal weights to gain the measure of similarity.

The pseudo-code to produce the MetricResults.TXT file is:

```
For cases =1 to 100
   For casen 1 to 100
     Component = Calculate-component-metric(cases, casen)
     Mcs = Calculate-component-metric(cases, casen) ; use CBRGraph
    Leaves = Calculate-component-metric(cases, casen) ; use ComponentCBR
    Cycles =Calculate-component-metric(cases, casen)
   Store results in MetricsResults.TXT
  end
end
```

ShapeCBR

ShapeCBR is a MS Access program that performs the CBR matching algorithm and is presented in Paper II in Appendix B. It is a prototype program and was used in the evaluation experiment. Given a target shape the program retrieves a set of closest matching cases.

Initially, the case base was set up by reading in the SHAPES.TXT file and storing the 100 shapes in a database table that represents the case base. The corresponding CAD file for each drawing was also read and associated with each case. Then the MetricResults.TXT file was read and stored in the MS Access table MetricResultsTable.

The CBR matching algorithm uses the following main function:

CBRMatch: The procedure takes a target case, weights, and searches the MetricsResults table, returning a table of cases and calculating a percentage of similarity. Pseudo-code for this CBR retrieval procedure is presented in Chapter 6 -Section 6.1.4.

Pseudo-Code to Find the Optimal Weights

The pseudo-code for determining the optimal weights is shown below. Further information about the weights and the variables used in the pseudo-code can be found in Chapter 6 - Section 6.4.2.

```
FUNCTION optimal-weights
 ITERATE through weights 0 TO 1 STEP 0.2
        FOR target = 1 TO case base size
            CBR-Retrieval(target, case base, weights) ; see Chpt.6-Section 6.1.4 for this algorithm
             ; take top retrieved result, i.e. the first record of result-table:
            retrieved-case = Result-Table(first record, retrieved Field)
            retrieved-percentage = Results-Table(first record, similarity percentage field)
            ; now fetch scores for that target and retrieved case:
            SEARCH(Score-Results-Table, target, retrieved-case) =>
                feeder-score, chill-score, orientation score, advice-score
        END
         Average Scores = mean(feeder-score, chill-score, orientation-score, advice-score)
        Store (weights, average scores) in Human-Comparison-Performance-Table
 END ITERATE
   SEARCH the Human-Comparison-Performance table for the best average scores & weights
   RETURN weights for best averages; these weights are optimal
END
```

Appendix G - Glossary of Casting Terms

This Appendix contains a glossary of casting terminology used in this thesis. For further terms, the reader is referred to MOD (1979).

Alloy: a mixture of two or more chemical elements at least one of which is a metal; for example, brass (which is a mixture of copper and zinc).

Blind feeders: there are a special type of feeder that work on the principle of using a 'fire cracker' core to puncture a hole in the steel shell that sets up as the blind feeder solidifies.

Casting: an object made by pouring molten metal into a **mould**.

Centreline Shrinkage: this is a type of internal shrinkage cavity typically affecting the central zones of extended parallel walled sections.

Chill: are heat-absorbing blocks and mouldable materials embedded in the mould that can force parts of the shape to freeze more quickly than neighbouring casting sections.

Cope: top half of a **mould**.

Core: separate refractory shape placed in mould cavity to allow the production of hollow sections.

Directional Solidification: the gradual completion of solidification towards a source of feed metal.

Drag: bottom half of a **mould**.

Feeding Distances: the maximum length that liquid metal can flow along a horizontal casting section towards a feeder(s) without freezing. Beyond the feeding distance, **Porosity** can occur.

Feeder: are reservoirs that can supply molten metal to elements of the shape to account for the volume loss as a result of shrinkage occurring as the casting alloy freezes from liquid to solid. See also **Risers**.

Foundry: a factory for casting metal.

Freezing Range: temperature range between **liquidus** and **solidus**.

Hot Spot: part of a casting that solidifies after the rest of the section freezes; resulting in **porosity** when the isolated spot freezes.

Investment Casting: a process in which a mould is produced by surrounding an expendable pattern with a refractory slurry that sets at room temperature. After this, the wax or plastic pattern is removed using heat prior to filling the mould with molten metal. When a wax mould is used, the process is called the *lost wax process*.

Liquidus: temperature at which molten alloy begins to solidify.

Methods Engineer: the task concerned with the design of the casting process; that is, filling the mould with molten metal, and the ensuing **solidification** of the metal. The Methods Engineer must consider a variety of problems associated with the process of solidification, the most crucial consideration being the **shrinkage** of metal that occurs as the metal freezes.

Methoding: the process of **Methods Engineering** that aims to ensure that a finished casting supplied to the customer is fault free and of the highest possible quality.

Modulus: the solidification time of a shape is proportional to the square of its volume to cooling surface area. This ratio is referred to as the geometric modulus, or just the modulus.

Mould: a shaped space into which molten metal is poured to solidify (see **Solidification**) into a required shape.

Pattern: a model of the required casting made from wood, metal or resin model constructed to size so that a **mould** can be formed of the required shape.

Pattern Makers Contraction Allowance: additional allowance made to drawing dimensions to compensate for metal contraction occurring after solidification; that is, during the process of **solidus** solidification.

Porosity: the reduction of volume that occurs during solidification can result in the formation of damaging cavities, known in the industry as *porosity*; from Greek *poros* meaning ‘passage’ or ‘pore’.

Riser: the term ‘riser’ is synonymous with the term ‘feeder’. Feeders are known as *risers* in America. See also **Feeders**.

Sand Casting: a technique of casting. Castings are formed this way in special sand containing a bonding agent in which a cavity corresponding to the outside shape of the **pattern** has been produced and into which the molten metal is poured and later solidifies.

Shrinkage: change in volume of metal between pouring temperature and **solidus**. Many problems, such as **porosity**, can occur during metal shrinkage.

Solidification: the process of metal becoming a solid.

Solidus: temperature at which solidification of alloy is complete.