RESEARCH NOTE

DEVELOPMENT OF AN INTELLIGENT CAVITY LAYOUT DESIGN SYSTEM FOR INJECTION MOLDING DIES

W. Hu and S. Masood

*Industrial Research Institute Swinburne (IRIS), Swinburne University of Technology
Hawthorn, Melbourne, Australia 3122*

(Received: October 29, 2001 – Accepted in Revised Form: July 28, 2002)

Abstract This paper presents the development of an Intelligent Cavity Layout Design System (ICLDS) for multiple cavity injection moulds. The system is intended to assist mould designers in cavity layout design at concept design stage. The complexities and principles of cavity layout design as well as various dependencies in injection mould design are introduced. The knowledge in cavity layout design is summarized and classified. The functionality, the overall structure and general process of ICLDS are explained. The paper also discusses such issues as knowledge representation and case-based reasoning used in the development of the system. The functionality of the system is illustrated with an example of cavity layout design problem.

Key Words Intelligent Design, Cavity Layout Design, Injection Mould Design, Case-Based Reasoning, Design Support System

چکیده در این مقاله توسعه سیستم هوشیار طراحی نقشه حفره برای قالبهای تزریق چند حفره ای گزارش می‌شود. هدف این سیستم، کمک به طراحان قالب در مرحله طراحی مفهومی است. پیچیدگیها و اصول طراحی حفره همراه با عوامل موثر بر طرح معرفی شده اند. دانش طراحی نقشه حفره طبقه بندی و فهرست شده است. قابلیت اجرا، ساختار کلی و فرایند عمومی سیستم تشریح شده و مباحثی از قبیل نحوه ارائه آگاهی و منطق مصداق محوری برای توسعه سیستم بحث شده است. قابلیت اجرای سیستم به کمک مثالی از یک مساله طراحی نقشه حفره بیان شده است.

1. INTRODUCTION

In manufacturing, the injection moulding is one of the most widely used production processes for producing plastic parts with high production rate and little or no finishing required on plastic products. The process consists of injecting molten plastic material from a hot chamber into a closed mould, allowing the plastic to cool and solidify and ejecting the finished product from the mould. For each new plastic product, the injection-moulding machine requires a new injection mould. Design and manufacture of injection mould is a time consuming and expensive process and traditionally requires highly skilled tool and mould makers. An injection mould consists of several components, which include mould base, cavities, guide pins, a sprue, runners, gates, cooling water channels, support plates, slides and ejector mechanism [1]. Design of mould is also affected by several other factors such as part geometry, mould material, parting line and number of cavities per mould.

With the advances in computer technology and artificial intelligence, efforts have been directed to reduce the cost and lead-time in the design and manufacture of an injection mould. Injection mould design has been the main area of research since it is a complex process involving several sub-designs related to various components of the mould, each requiring expert knowledge and experience. Mould design also affects the productivity, mould maintenance cost, manufacturability of mould, and the quality of the moulded part. Most of the work in mould design has been directed to the application of expert systems, knowledge based systems and artificial intelligence to eliminate or supplement the vast amount of human expertise required in traditional design process. Kruth and Willems [2] developed an intelligent support system for the design of injection moulds integrating commercial CAD/CAM, a relational database and an expert system. Lee et al [3] proposed a systematic methodology and knowledge base for injection mould design in a concurrent engineering environment. Raviwongse and Allada [4] developed a neural network-based design support tool to compute the mould complexity index to help mould designers to assess their proposed mould design on mould manufacturability. Kwong and Smith [5] developed a computational system for the process design of injection moulding based on the blackboard-based expert system and the case-based reasoning approach, which includes mould design, production scheduling, cost estimation and determination of injection moulding parameters.

Several studies have also been made on improving the design of specific components of an injection mould. Ong et al [6] developed a knowledge-based and object-oriented approach for the design of the feed system for injection moulds, which can efficiently design the type, location and size of a gating system in the mould. Irani et al [7] also developed a software system for automatic design of gating and runner systems for injection moulds and provides evaluation of gating design based on specified performance parameters. Nee et al [8] proposed a methodology for determination of optimal parting directions in injection mould design based on automatic recognition and extraction of undercut features. Chen and Chou [9] developed algorithms for selecting a parting line in mould design by computing the undercut volumes and minimising the number of undercuts. Park and Kwon [10] worked on the design of cooling systems in injection moulds and proposed an optimal design based on thermal analysis and design sensitivity analysis of the cooling stage of the injection moulding process.

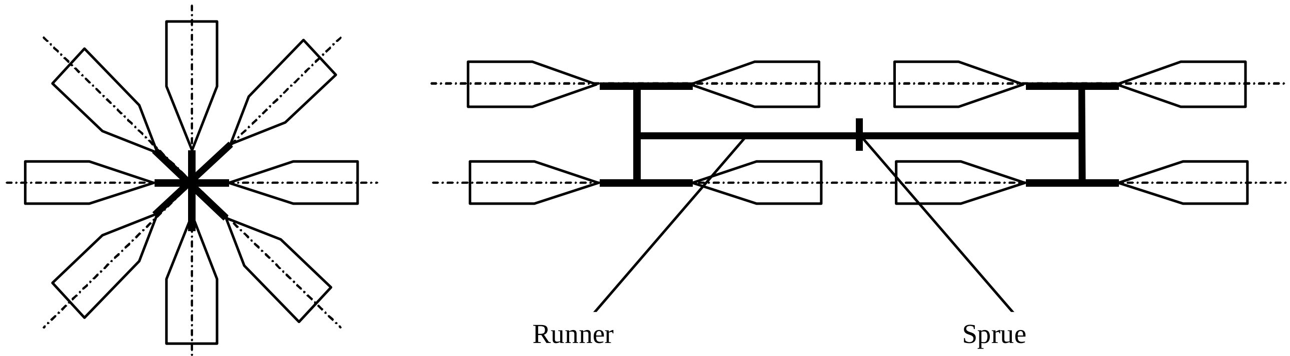
One area in injection mould design, which has received little attention, is the design of cavity layout in a multiple cavity injection mould. Cavity layout design affects the whole process of injection moulding directly, since it is one of the most important phases in mould design process. Consideration of cavity layout design in injection mould at concept design stage will improve the quality of injection moulded products because it is associated with the determination of many key factors affecting the design and quality of mould. Such factors include number of cavities; parting line; type of mould; type and position of gate; runner system; cooling system and ejection system. Some of these factors are difficult to build as true mathematical models for analysis and design.

This paper presents the development of a design support system, called Intelligent Cavity Layout Design System (ICLDS), for multiple-cavity injection moulds based on knowledge based and object oriented approaches. It uses the case-based and ruled-based reasoning in arriving at the layout solution [11]. It is based on the commercial software system named "RETE++", which is an integrated development platform for customers to develop their own knowledge-based systems [12]. The objective is to make full use of available techniques in artificial intelligence in assisting mould designers at concept design stage.

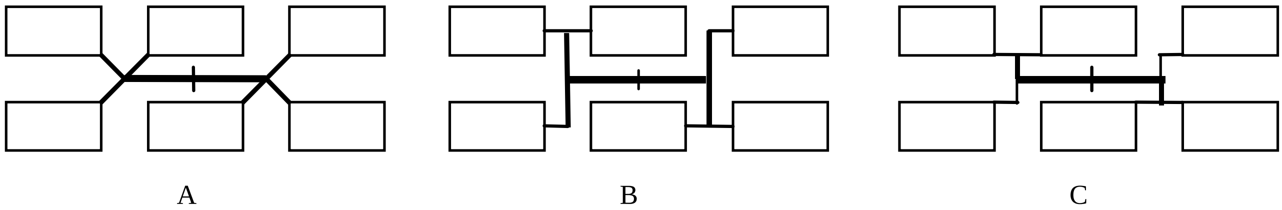
2. CAVITY LAYOUT DESIGN IN INJECTION MOULDS

Current practice for injection mould design, especially cavity layout design, depends largely on designers' experiences and knowledge. It would therefore be desirable to use knowledge engineering, artificial intelligence and intelligent design techniques in generating an acceptable cavity layout design in injection mould accurately and efficiently. In mould design, most of patterns of cavity layout and rules and principles of cavity layout design can also be easily represented in the form of knowledge, which can be used in most of knowledge-based design systems.

For example, for the layout patterns shown in Figure 1, the criteria to select the suitable layout pattern for design are mainly dependent on working environments, conditions and requirements of customer. To make a choice of contradictory factors will rely obviously on designer's knowledge and experiences. It is rather suitable for intelligent design techniques to be used in systems designed for such situations, especially for routine or innovation



(a) Examples of 8-cavity mould layouts in a circular pattern and a rectangular pattern



(b) Examples of variations of rectangular 6-cavity layouts:
A. Y-style runners,
B. B. H-style runners with two changes in direction of plastic flow, and
C. C. H-style runners with three changes in direction of plastic flow.

Figure 1. Some patterns of cavity layout with multiple cavities.

design.

Design of injection mould mainly involves consideration of design of the following elements or sub-systems:

(1) mould type
(2) number of cavities
(3) cavity layout
(4) runner system
(5) ejector system
(6) cooling system
(7) venting
(8) mounting mechanism

Most of the elements are inter-dependent such that it is virtually impossible to produce a meaningful flow chart covering the whole mould design process. Some of the design activities form a complicated design network as shown in Figure 2.

Obviously, in injection mould design, it is difficult for designer to monitor all design parameters. Cavity design and layout directly affects most of other activities. The application of advanced knowledge based techniques to assist designer in cavity layout design at concept design stage will greatly assist in the development of a comprehensive computer-aided injection mould design and manufacturing system.

It is noted from Figure 1 that a number of different layout patterns are possible with multiple cavities inside a mould. Higher the number of cavities of the mould results in higher productivity of the injection mould. But this may lead to difficulties with issues such as balancing the runners or products with the complicated cavity shapes, which in turn may lead to problems of mould manufacturability. It is also possible that the

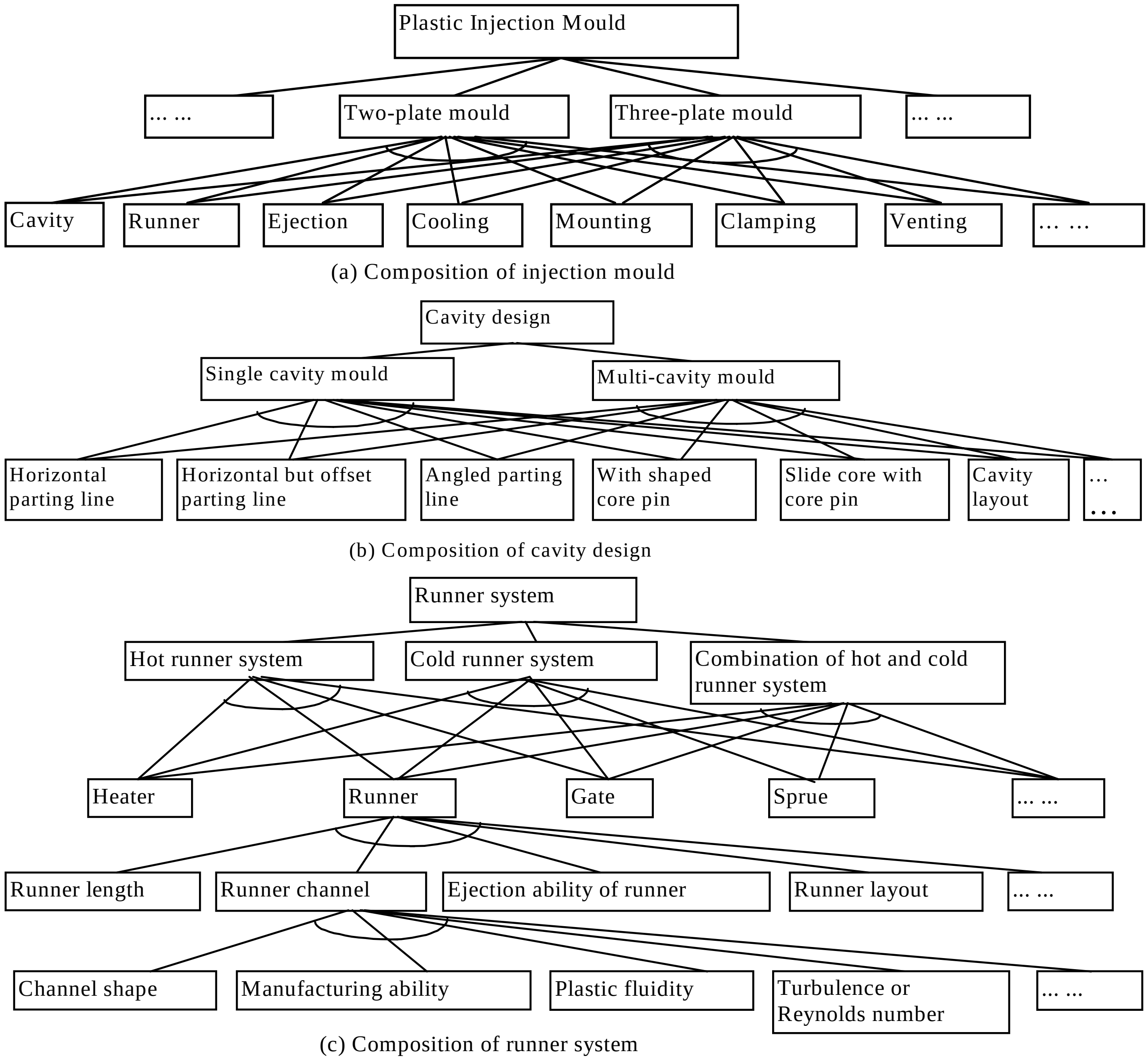


Figure 2. Design network of injection mould.

number of cavities and the pattern of cavity layout will influence the determination of parting line, type of gate, position of gate, runner system and cooling system. Most of the main activities of mould design are therefore linked to cavity layout design. Figure 3 shows the relations between cavity layout design and other design activities. The cavity layout design problem therefore depends upon a number of functionalities of the overall mould design system, which includes:

(1) definition of design specifications including analysis and description of characteristics of design problem

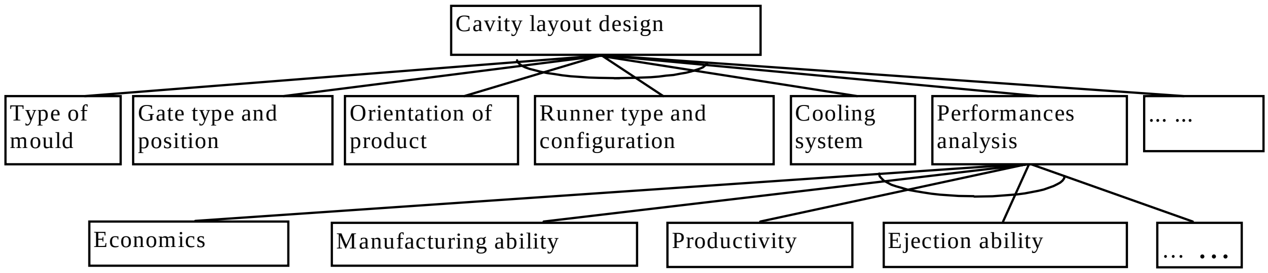


Figure 3. Relationship diagram between cavity layout design and other modules of mould.

(2) determination of mould type
(3) determination of number of cavities
(4) determination of orientation of product
(5) determination of runner type and runner configuration
(6) determination of type and position of gate
(7) cavity layout conceptual design
(8) evaluation of ejection ability, manufacturing ability and economic performances
(9) determination of cooling system
(10) graphic results display and output

3. STRUCTURE OF ICLDS AND THE DESIGN PROCESS

The structure of the Intelligent Cavity Layout Design System (ICLDS) is based on case-based reasoning and ruled-based reasoning designed around the RETE++ software system. Figure 4 shows the overall structure of ICLDS schematically. Figure 5 shows the general design process of ICLDS. The design process starts with the definition of design specifications. The ICLDS system retrieves similar cases from case base by computing the similarity between the cases and the new case. If the solution is satisfactory, then results are displayed graphically. If the solution is not satisfactory, then ICLDS will use rule-based reasoning with forward or backward chaining or a mixture of both to arrive at a solution. If the solution is still unsatisfactory, then the user has to modify some of the initial design specifications. The use of case-based technology in the design process in ICLDS allows the user to obtain the solution(s) of design problem more quickly and flexibly.

The structure of knowledge base and database used in the development of ICLDS is based on the underlying knowledge base and database structure from the RETE++ software system, which is a commercially available software development platform.

4. DEVELOPMENT OF ICLDS

4.1 Classifications of Knowledge For various logic and steps involved in layout design, there are different kinds of knowledge that needs to be described and represented in cavity layout design. The types of knowledge can be classified into five kinds based on object oriented (OO) concept as described below:

(1) Design instance/case: previous design cases and current design instances
(2) Relation: superclass—class—subclass relation, class—instance relation
(3) Attribute: design variables, features, attributes of design problem
(4) Rule: general design rules, design experiences
(5) Procedure and/or model: numeric calculation, mathematical modeling, analysis, evaluation and procedures.

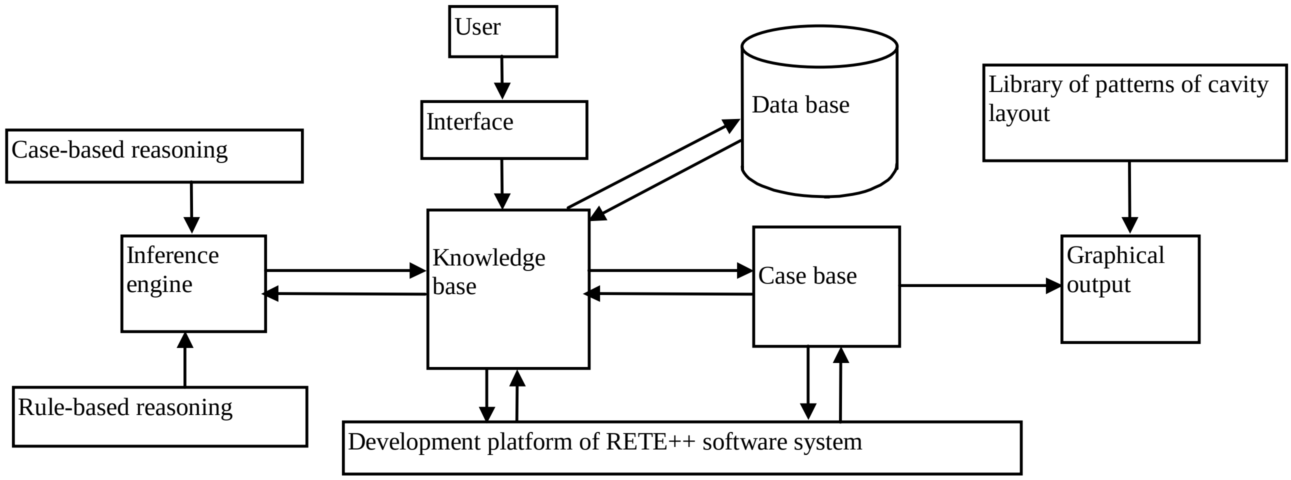


Figure 4. Overall structure of ICLDS.

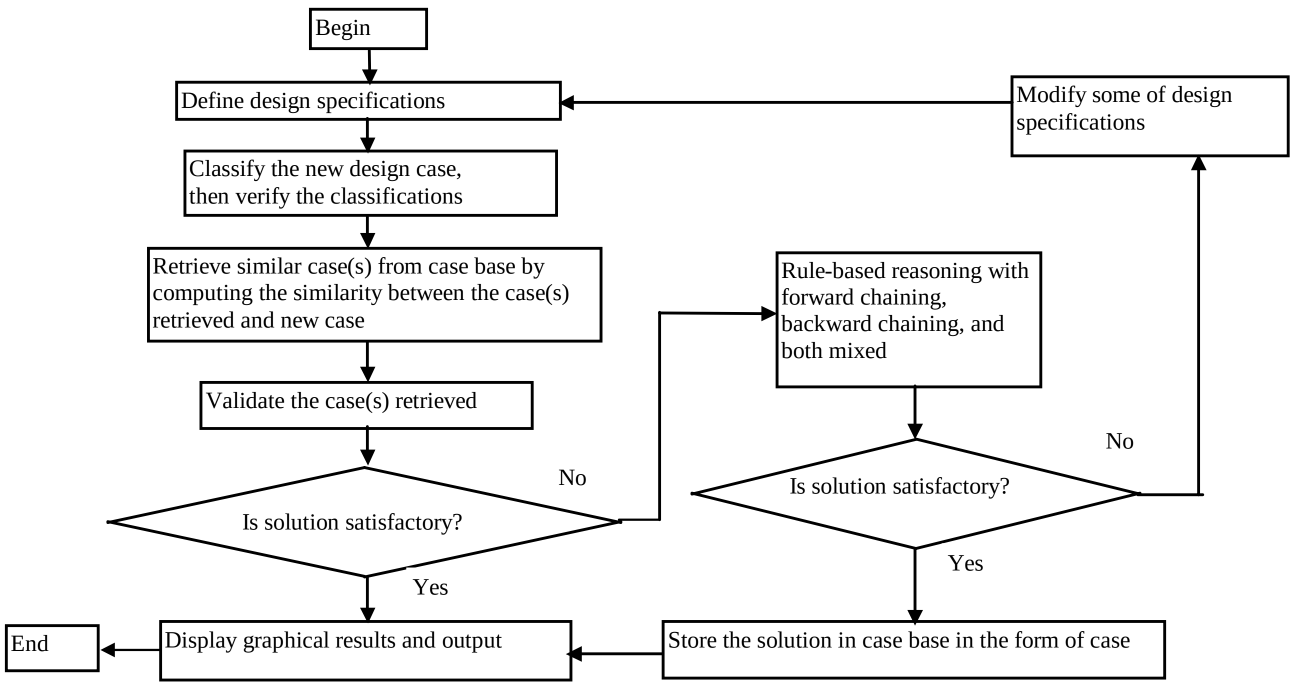


Figure 5. The general design process of ICLDS.

4.2 Knowledge Representations To describe each of these types of knowledge, the internal data structures of the ECLIPSE language, included in RETE++ inherently, can be used to make the object orientated representation of the design process as explained earlier. Some other considerations in knowledge representation are as follows:

(1) For “design instance/case”, we combine “fact definition” and “relation definition” plus database and case base to represent it

(2) The “attribute” are represented as instances of “template definition” and/or “relation definition”

(3) For “relation”, we use “relation definition” to describe it

(4) For “rule”, we combine “rule definition” and “rule set definition” to represent it

(5) The “procedure/model” are defined by external routines using C++ language

Furthermore, “goal definition” and “goal generation” techniques are used to fulfill backward chaining reasoning, and “case-based reasoning” is used to carry out case-based design.

4.3 Case-Based Reasoning Case-Based Reasoning (CBR) is dependent firstly on case retrieved. Case-based retrieval is based on “Similarity Metric”. Therefore, how to calculate the similarity is obviously the key technique in CBR, and it is described in detail as below.

Similarity metric is a weighted distance function in a multi-dimensional space where each dimension corresponds to a field whose value is specified in the query (new case) and which has a non-missing value in the case being ranked. The distance between the case and the query (which corresponds to a point in this multi-dimensional space) is computed differently for ordinal and nominal fields. An ordinal field is a field whose values are ordered or sorted. A nominal field is one whose values represent qualitative information for which sorting makes no sense. In general, ordinal fields include dates, integers, and real numbers while nominal fields include Boolean, Symbols, and Text.

(1) Ordinal Distance

For ordinal fields the distance, d_{ij}, between the i-th case’s value, v_{ij}, and the query’s value, V_j, for the j-th field, is computed as:

$$d_{ij} = \frac{\left|v_{ij} - V_j\right|}{\left|V_j^{\max} - V_j^{\min}\right|} \tag{1}$$

where the maximum and minimum values for each field are determined during index construction.

Since d_{ij} represents the distance along the j-th axis of an n-dimensional similarity space, the similarity space distance D_i is given by:

$$D_i = \frac{\sqrt{\sum_{j=1}^{N} d_{ij}^2}}{N} \tag{2}$$

which, since d_{ij} must range between 0 and 1, must also range between 0 and 1. When weights are used, the above equation becomes:

$$D_i = \frac{\sqrt{\sum_{j=1}^{N} (W_j\, d_{ij})^2}}{\sum_{j=1}^{N} W_j} \tag{3}$$

which, since W_j and d_{ij} must range between 0 and 1, must also range between 0 and 1.

(2) Nominal Distance for Text

To determine the distance between the value of a TEXT field in a case and that specified in a query, we determine a weight for each term by which a text field is indexed and a weight for each term in the query. These weights are computed according to the following formula, where:

$$W_{ik} = \frac{F_{ik}\, Log(n_k / N)}{\sqrt{\sum_{j=1}^{N} (F_{ij}\, Log(n_j / N))^2}} \tag{4}$$

N is the number of cases.
n_k is the number of different cases in which term k occurs.
F_{ik} is the number of occurrences of term k in case i divided by the total number of terms in case i.

W_{ik} is the weight of the k-th term in case i.

Let W_k be the weight of the k-th term in the query, computed as in the above formula.

Let T be the number of terms in the query.

Given these weights, the similarity (expressed as a normalized distance) between two text fields is computed as:

$$S_i = \frac{\sum_{k=1}^{T} W_{ik} W_k}{\sqrt{\sum_{k=1}^{T} (W_{ik})^2 \sum_{k=1}^{T} (W_k)^2}} \quad (5)$$

(3) Nominal Distance for Symbols

Symbols are merely a special case of text with only one term. The weights for symbol fields are computed as in the above equations for text fields with F_{ik} always being one. Given these weights, the similarity is computed exactly as for text fields.

4.4 Validation of Case Validation of case is to check up whether each acceptable case is suitable for current problem and to find out the most suitable one, so each case should be associated with testing methods and tested results on it. Only the case, under the given conditions, for which all tested results on it match those of the current design problem, can be considered as the solution prototype for further refining.

4.5 Criteria for Validity of Cost Reduction With the application of ICLDS for cavity layout, two kinds of cost reduction can be expected. One is the overall theoretical cost reduction achieved in using the system to carry out the conceptual design of injection moulds. The other is the practical cost reduction value recorded in the case base, which may be used to do the case-base reasoning if the case has the "cost reduction" attribute. For the theoretical one, there is no need of any criteria for validity of cost reduction because the cost savings will obviously come out through lead-time saving, improvement in design quality and quick response to customers. For the validity of practical cost reduction, the criterion of comparison can be used. For example, we can compare the "cost reduction" attributes for two cases and determine which one provides better fit for the customer's requirements. One can use the percentage cost reduction formula to do the comparison. The percentage cost reduction can be calculated by the formula:

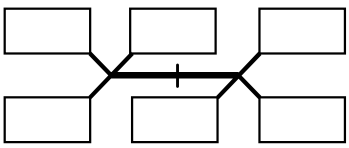


Figure 6. Graphic result of pattern of 6-cavity layout----“Y-style runners”.

Cost Reduction = (previous cost - current cost)/previous cost

It is better first to work out the percentage for the cost reduction attributes and then do the validation of case between the cases.

5. EXAMPLE OF APPLICATION

An application example, “determination of cavity layout pattern” of the “conceptual design for cavity layout” provided by Intelligent Cavity Layout Design System (ICLDS) is given below:

If the initial design conditions are:

(1) What type of mould is used? Two plate

(2) What type of runner is used? Cold runner

(3) How many cavities are there in mould? 6

(4) How long is it required for product to clear the moulding area? Small

(5) What shape of product does moulding make? Rectangle

Then the result is given by: (this is shown in Figure 6).

Pattern of cavity layout design is: <u>Y–Rectangular-Layout</u>

The knowledge base is developed using features of ECLIPSE language, such as ‘defrelation’, ‘deftemplate’, ‘defruleset’, and ‘goal’ generation. Part of the program, which describes the overall format of knowledge base development, is listed below:

```
………………
(defrelation dimension (?item ?size))
(defrelation layout (?item ?type))
………………
 (defruleset Runner_system    10 (agenda body))
………………
(defruleset cavity_layout 8
   (agenda body 2)
………………
(defruleset cavity_layout (agenda body))
 (defrule goal_cavity_layout
  (initial-fact)
  (goal (selection layout_designed ?yesno))
  (unknown (selection layout_designed ?yesno))
  (not (layout cavity ?type))
  =>
  (printout t "... Waiting! Waiting! Waiting! ..." t))
……………
(defrule cavity_layout2_2_01
  (goal (layout cavity ?type))
  (unknown (layout cavity ?type))
  (known (quantity number_of_plate 2))
  (known (quality type_of_runner "cold_runner"))
  (known (quantity number_of_cavity 2))
  (known (quality clear_time "small"))
  =>
  (assert (layout cavity "horizontal_layout")))
……………
(defrule cavity_layout2_6_01
  (goal (layout cavity ?type))
  (unknown (layout cavity ?type))
  (known (quantity number_of_plate 2))
  (known (quality type_of_runner "cold_runner"))
  (known (quantity number_of_cavity 6))
  (OR (whatis shape_of_product ~"elongated")
   (whatis size_of_product ~"small"))
  =>
  (assert (layout cavity "Y_rectangular_layout")))
……………
```

6. CONCLUSION

The problem of design of cavity layout in multiple cavity injection moulds has received relatively little attention in computer based design support systems for injection molding. The development of Intelligent Cavity Layout Design System (ICLDS) is believed to be the first attempt in this direction using knowledge-based approach. The development of ICLDS for injection mould is based on RETE++ in Windows environment on PC. From a practical point of view, ICLDS can be used as a tool for designer to implement cavity layout design of injection mould at concept design stage. It provides a positive step towards the development of a fully automated injection mould design process from product model to mould manufacturing.

7. REFERENCES

1. Menges, G. et al., “How to Make Injection Molds”, Hanser Publisher, Munich, (1986).
2. Kruth, J. P. and Willems, R., “Intelligent Support System for the Design of Injection Moulds”, ***Journal of Engineering Design***, 4(5), (1994), 339-351.
3. Lee, R. S., Chen, Y. M. and Lee, C. Z., “Development of a Concurrent Mould Design System: a Knowledge Based Approach”, ***Computer Integrated Manufacturing Systems***, 10(4), (1997), 287-307.
4. Raviwongse, R. and Allada, V., “Artificial Neural Network Based Model for Computation of Injection Mould Complexity”, ***International Journal of Advanced Manufacturing Technology***, 13(8), (1997), 577-586.
5. Kwong, C. K. and Smith, G. F., “A Computational System for Process Design of Injection Moulding: Combining Blackboard-Based Expert System and Case-Based Reasoning Approach”, ***International Journal of Advanced Manufacturing Technology***, 14(4), (1998), 239-246.
6. Ong, S. K., Prombanpong, S. and Lee, K. S., “An Object-Oriented Approach to Computer-Aided Design of a Plastic Injection Mould”, ***Journal of Intelligent Manufacturing***, 6(1), (1995), 1-10.
7. Irani, R. K., Kim, B. H. and Dixon, J. R., “Towards

Automated Design of the Feed System of Injection Moulds by Integrating CAE, Iterative Redesign and Features", ***Transactions ASME Journal Engineering for Industry***, 117(1), (1995), 72-77.

8. Nee, A. Y. C., Fu, M. W. et al, "Determination of Optimal Parting Directions in Plastic Injection Mould Design", ***Annals CIRP***, 46(1), (1997), 429-432.
9. Chen, L. L. and Chou, S. Y., "Partial Visibility for Selecting a Parting Direction in Mould and Die Design", ***Journal of Manufacturing Systems***, 14(5), (1995), 319-330.
10. Park, S. J. and Kwon, T. H., "Thermal and Design Sensitivity Analyses for Cooling System of Injection Mould. Part 2:Design Sensitivity Analysis", ***Transactions ASME Journal Manufacturing Science & Engineering***, 120(2), (1998), 296-305.
11. Maher, M. L. et al., "Developing Case-Based Reasoning for Structural Design", ***Intelligent System and Their Applications***, IEEE Expert, USA, (June 1996).
12. The Haley Enterprise, Inc., "Documentation of RETE++", (1994).